(12) LEVEL III

AD

AD-E400 645

ADA101357

TECHNICAL REPORT ARLCD-TR-80041

AN ANTI-RADIATION PROJECTILE (ARP) TERMINAL EFFECTS SIMULATION COMPUTER PROGRAM (ARPSIM)

R. D. WEBSTER

JUNE 1981

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
LARGE CALIBER
WEAPON SYSTEMS LABORATORY
DOVER, NEW JERSEY

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

81 6 29 024

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: TECHNICAL REPORT ARLCD-TR-80041 ✓	2. GOVT ACCESSION NO.: AD-A101 357
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): AN ANTI-RADIATION PROJECTILE (ARP) TERMINAL EFFECTS SIMULATION COMPUTER PROGRAM (ARPSIM)	5. TYPE OF REPORT & PERIOD COVERED: Final
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): R. D. Webster	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: ARRADCOM, LCWSL Systems Development and Modelling Div (DRDAR-LCS) Dover, NJ 07801	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: ARRADCOM, TSD STINFO Div (DRDAR-TSS) Dover, NJ 07801	12. REPORT DATE: June 1981 13. NUMBER OF PAGES: 105
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Guidance accuracy	Direct hit effect
Fuzing schemes	Radar blast
Homing projectile	Vehicle body blast
Monte Carlo simulation	Linear trajectories
Warhead fragmentation effects	Radar emitting target

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is the documentation for a computer code developed primarily to aid development engineers by providing estimates of the relative importance of components in terms of effectiveness.

The ARPSIM computer model was developed in support of a requirement to estimate the effectiveness of the various kill mechanisms (fragmentation, antenna blast, vehicle blast, and direct hit) of an Anti-Radiation Projectile (ARP) against a typical air defense radar-emitting target. A Monte Carlo technique

(cont)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

is used to generate estimates of the probability of kill for a single ARP fired against a single target. The influence of various fuzing schemes and guidance errors are considered in determining burst points.

CONTENTS

FIGURES

INTRODUCTION

ARPSIM is a computer program developed to provide estimates of the terminal effectiveness of an Anti-Radiation Projectile (ARP) fired against an air defense, radar-emitting target.

The primary objective of ARPSIM is to provide the user with a tool to parametrically ascertain the sensitivity of the ARP to warhead, guidance, and fuzing design changes.

The ARPSIM model simulates single round terminal conditions from the time when the ARP is flying a straight line trajectory at some fixed attack elevation in the vicinity of the target. Trajectories are determined from guidance errors distributed about a specified homing point. No further trajectory alterations are made. Fuzing points on the target are specified, and when fuzing conditions are satisfied, a burst point is established along the selected trajectory. The proximity of the burst point to the target determines the magnitude of kill probabilities for blast, direct hit, and fragmentation effects. Separate blast kills for both the target body and radar antenna can be estimated. Fragmentation effects are based upon terminal effectiveness estimates generated by the full spray material lethal area (MAE) computer code (refs 1 and 2).

The ARPSIM program is coded in FORTRAN for interactive use on the CDC 6500/6600 in the INTERCOM mode. The user is prompted for data entry. Also, at the option of the user, an input guide can be generated prior to each use. Fragmentation effects are estimated from data previously generated by the MAE program relative to conditional kill probabilities. Optionally, a function, $P_k(r)$, can be provided to specify fragmentation kill probability as a function of range. Comments are added throughout the FORTRAN code for better understanding and for development of future options for the code.

A user guide, an example of a computer run, and a FORTRAN code listing are presented as appendixes A, B, and C.

PROGRAM FLOW

For each Monte Carlo sample, a simulation of the terminal characteristics of the ARP is made beginning at a time prior to fuzing during the ARP flight after final corrections to the trajectory have been made and when the remaining trajectory is linear at a fixed attack angle. The sequence of events for each simulation is:

1. An attack angle is chosen which provides a straight line flight path with respect to a specified homing point.

2. A trajectory is chosen based upon the guidance errors with respect to the homing point.

3. A fuzing point along the chosen flight path is determined.

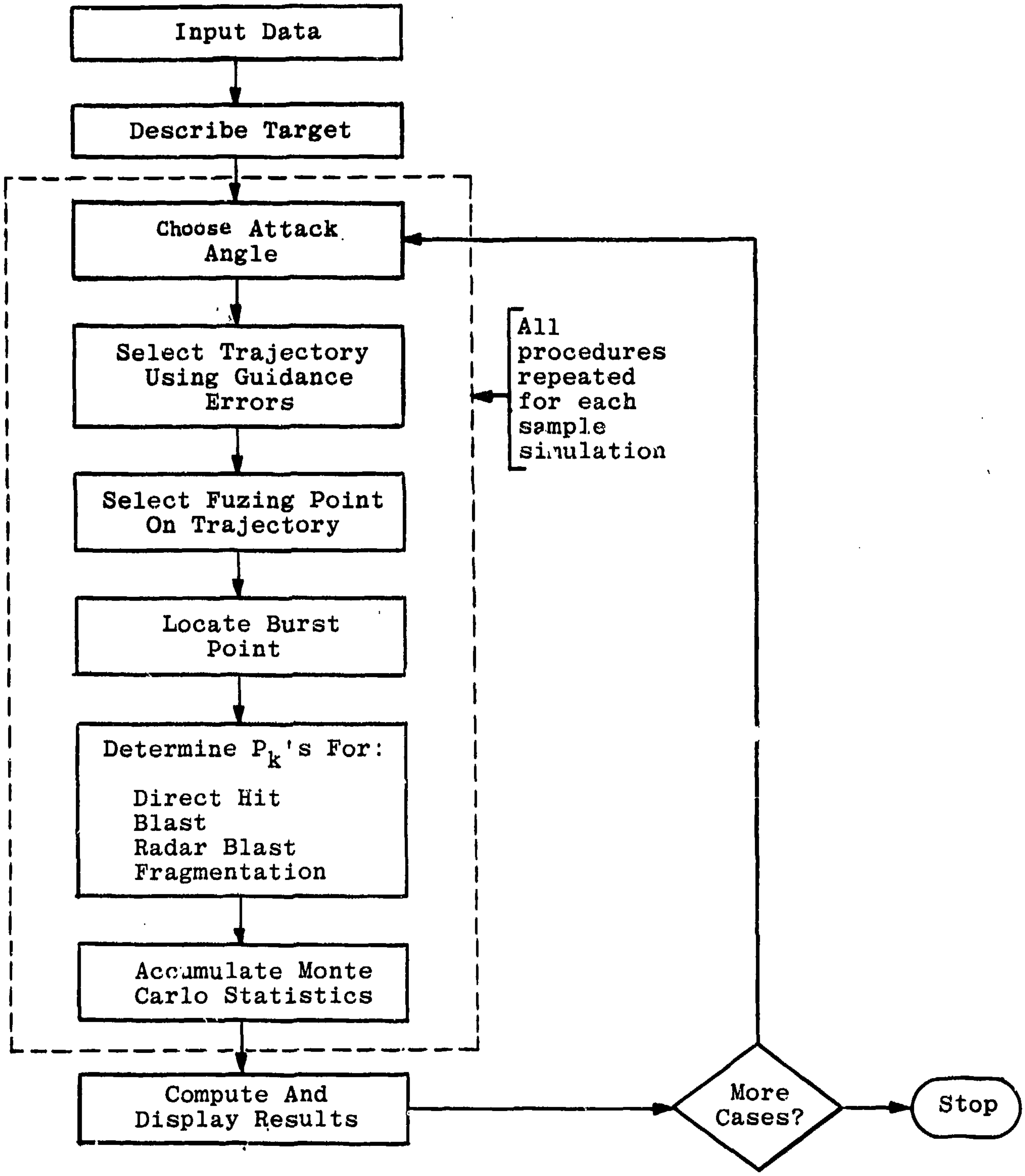

Figure 1. Program flow

4. A burst point is established based on the type of fuze, direct hit potential and possible backup fuzing or ground impact prior to nominal fuzing.

The proximity of the burst point to the target yields estimates of kill probability for direct hit, target body blast, radar blast, and fragmentation effects. The overall kill probability for each simulation is determined from the individual kill mechanism effects. This process is repeated for each simulation to provide the desired estimates of ARP terminal effectiveness. The above-described program flow is illustrated in figure 1.

The following subsections briefly describe portions of the model in the approximate order in which they follow the program flow.

Terminal Effects

Terminal effects are measured in terms of direct hit, blast, and fragmentation. Knowledge of the ARP warhead characteristics as well as the target's vulnerability to each of these effects is essential. Consequently, a preliminary analysis is required of the vulnerability of the target to the ARP warhead. Fragmentation effects are provided in either of two distinct formats: a P_k grid which yields conditional kill probability as a function of burst point proximity to the target, or a P_k vs R (range) function which provides the kill probability data as a function of range only; i.e., azimuth characteristics are averaged for each range from projectile burst to target. These functions are provided by the MAE program. Direct hit and blast effects are estimated from standard target vulnerability analysis.

The overall kill probability for each Monte Carlo sample is based upon these individual effects and is computed as:

$$P_k = 1-(1-P_{DH})(1-P_{RDR})(1-P_{BLST})(1-P_F)$$

where P_{DH} = direct hit kill probability,

P_{RDR} = radar blast kill probability,

P_{BLST} = vehicle blast kill probability,

and P_F = fragmentation kill probability.

Coordinate System

The simulation uses a rectangular coordinate system whose origin is at ground zero of the target center of vulnerability. Target heading establishes the negative range direction (-R); positive deflection (D) is to the left (driver's side) of the target; height (H) is measured from the ground (positive up) (fig. 2).

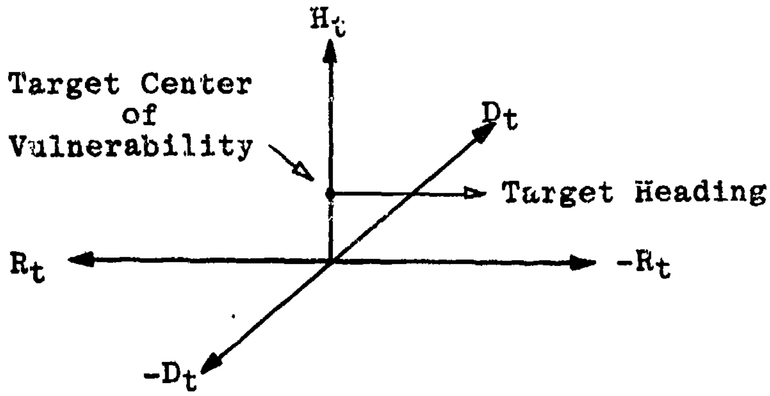

Figure 2. Coordinate system

Attack Angle

The attack angle is the combination of both elevation and azimuth angles which define the direction of the incoming ARP with respect to the coordinate system for the target. Azimuth is measured from the negative range direction toward the positive deflection. The elevation angle, ω, is measured from the horizontal in the positive height direction (fig. 3). Azimuth can be either fixed or chosen randomly for each simulation. Elevation is chosen from a Gaussian distribution with a specified mean, μ_ω, and standard deviation, σ_ω. The attack angle orients the direction of the ARP flight path (trajectory).

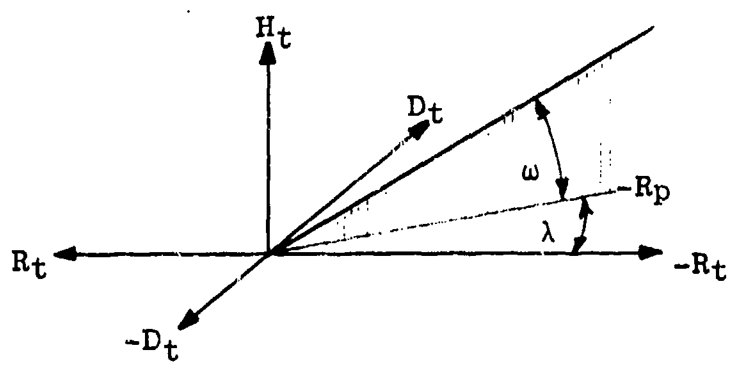

Figure 3. Attack angle

Guidance Errors

Guidance errors are Gaussian and are specified by either the standard deviations in deflection and height or CEP in deflection and height. These errors are defined in the plane normal to the ARP trajectory and passing through the homing point. The location of the guidance plane and the selection of a sample trajectory through the point (GR, GD, GH) are illustrated in figure 4. The determination of the point (GR, GD, GH) is as follows:

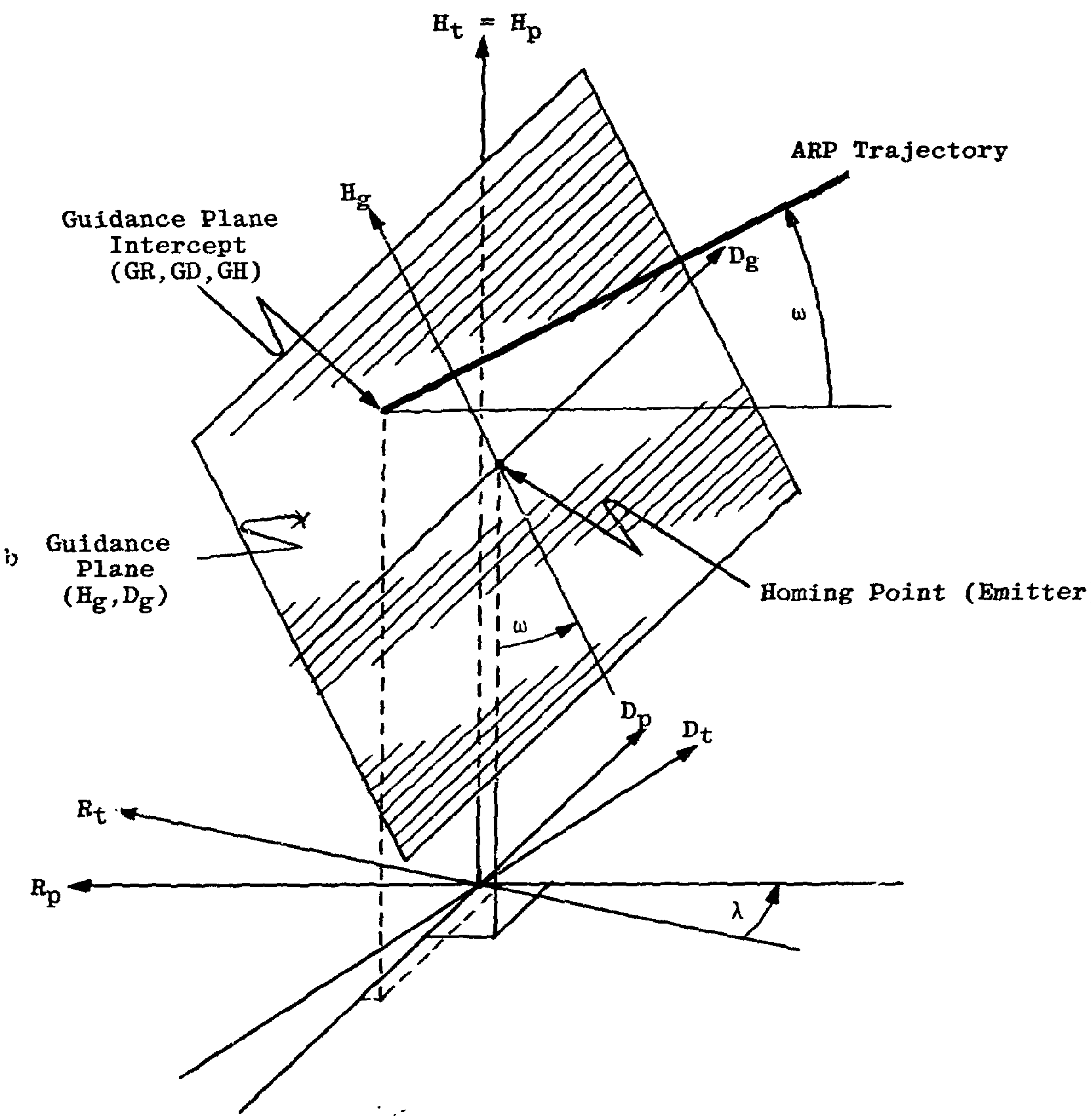

Figure 4. *Guidance errors*

First, the homing point (GMR, GMD, GMH), defined in the target coordinate system (R_t, D_t, H_t), is rotated through the azimuth angle, λ.

$$GMR' = GMR*\cos(\lambda) - GMD*\sin(\lambda)$$
$$GMD' = GMD*\cos(\lambda) + GMR*\sin(\lambda)$$

Then GR, GD, and GH are defined based on the sampled errors about the rotated homing point. Then

where H, D are random normal deviates with $\mu = 0$, $\sigma = 1$,

$$GR = GMR' + H*\sigma_h*\sin(\omega)$$
$$GD = GMD' + D*\sigma_d$$

and,
$$GH = GMH + H*\sigma_h*\cos(\omega)$$

where GR, GD, GH are in the R_p, D_p, H_p (projectile) coordinate system and σ_h, σ_d are the standard deviations in height and deflection, respectively, of the guidance error in the guidance plane (H_g, D_g).

Fuzing

Six options are available for primary fuzing; both point detonating (PD) and proximity (VT) backup fuzes can be considered. Each of the primary fuzes is described below:

Gaussian Fuzing Angle

Fuze glitter points are specified on the target and a single glitter point is selected as either the first glitter point encountered or, optionally, chosen randomly for each simulation. When the angle between the flight path and a line from the ARP to the glitter point is equal to the fuzing angle, ϕ, the point on the trajectory at the vertex of the angle is taken to be the fuzing point (fig. 5). The fuze angle for each simulation is selected from a Gaussian distribution as,

$$\phi = \mu_\phi = \nu*\sigma_\phi$$

where ν is a random normal deviate with $\mu = 0$ and $\sigma = 1$.

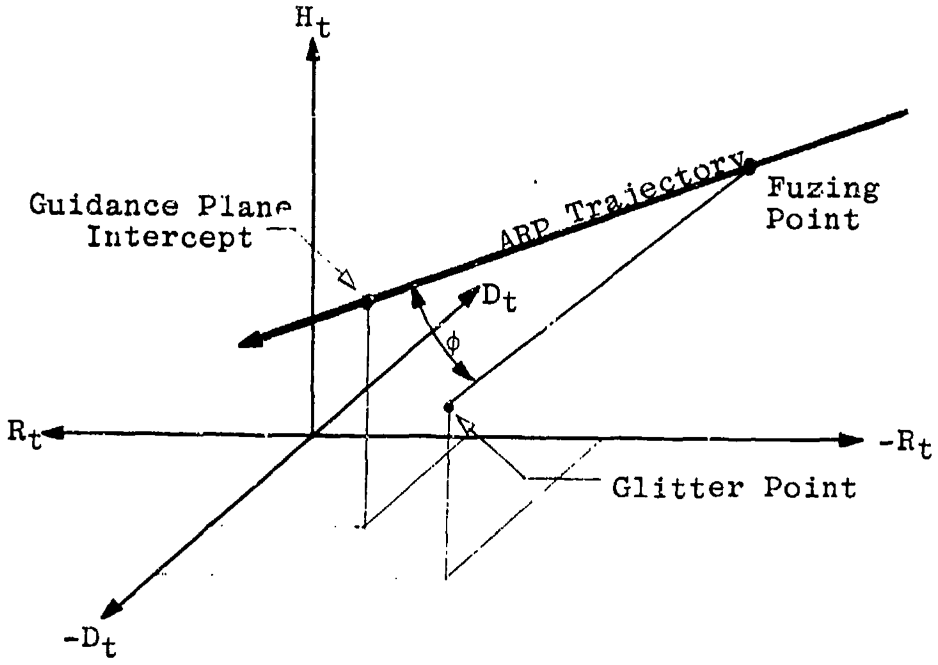

Figure 5. Fuzing angle

Uniform Fuzing Angle

Identical to the Gaussian fuzing angle except that ϕ is chosen as uniformly random between specified limits for each simulation.

Linear Fuzing

Fuzing occurs at some distance along the ARP flight path measured from the guidance plane. The distance along the flight path is chosen from a Gaussian distribution with a specified mean, μ_1 and standard deviation, σ_1 (fig. 6). Given the ARP terminal velocity, linear fuzing can be used to represent a time fuze where time is measured from the guidance plane. If μ_t, σ_t represent the Gaussian parameters for a time fuze, then where V_T is the ARP terminal velocity, $\mu_1 = V_T*\mu_t$ and $\sigma_t = V_T*\sigma_t$.

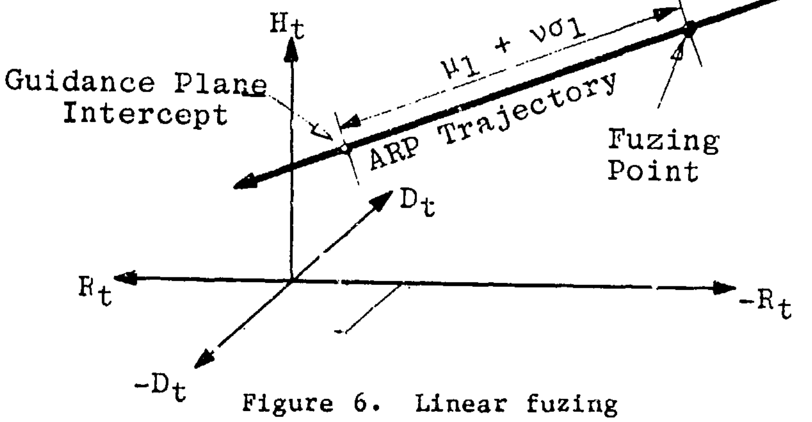

Figure 6. Linear fuzing

Height Fuzing

Fuzing occurs at a specific height above the ground. Height is chosen from a Gaussian distribution where the mean and standard deviation are specified. The point on the ARP flight path which corresponds to the selected height is the fuzing point.

VT Fuze

A VT fuze functioning distribution is considered by specifying the cumulative distribution function of fuzing height. A fuzing height is chosen according to sampling from that distribution and the fuzing point is the point on the ARP flight path which corresponds to the selected height.

PD Fuze

The intersection of the flight path with the ground establishes the PD fuzing point.

All of the above described primary fuze options can have either a PD or VT backup fuze. The backup fuze is used if a test for primary fuze functioning fails; otherwise, the primary fuze establishes the fuzing point unless a VT backup fuze point occurs at a greater height than the height component of the primary fuze point.

Target

The physical dimensions of the target are represented by a group (up to 5) of rectangular parallelepipeds (fig. 7) with the center of target vulnerability located over the origin of the ARP terminal coordinate system (R_t, D_t, H_t).

Burst Point

In all cases, once the fuzing point is found, a check is made to ascertain whether the target has been penetrated in order to reach that fuze point. If such penetration is found, the first penetration point is taken as the warhead functioning burst point (in this case, a direct hit burst point). Since the burst point is established in the rotated coordinate system (through the azimuth component of the attack angle), prior to determining kill effects, the burst point is rotated back into the target coordinate system.

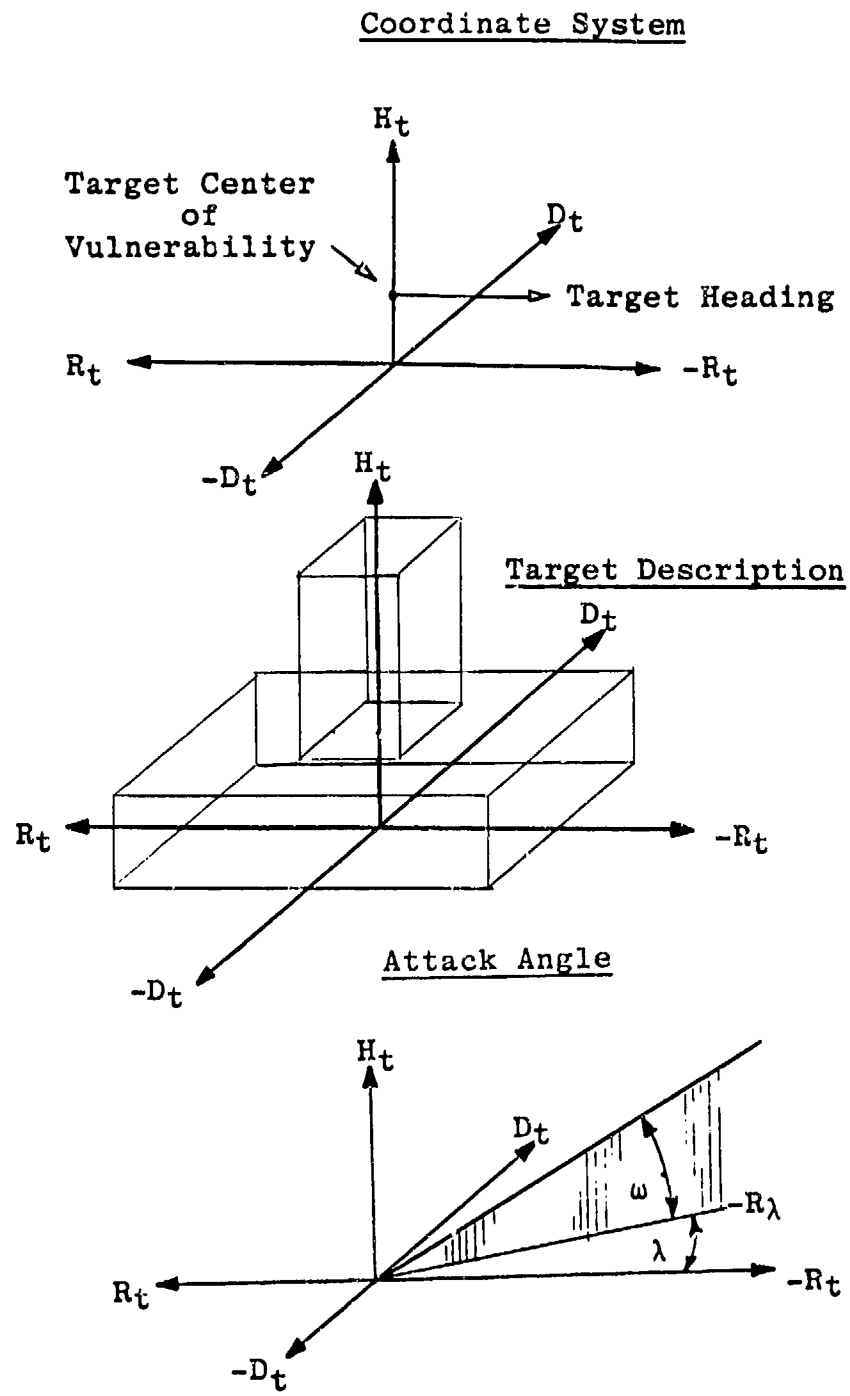

Figure 7. Target description

Direct Hit

If the burst point of the ARP is found to be at the surface of a parallelepiped representing a face of the target, a direct hit is deemed to have occurred.

Blast

Blast kills can be estimated for both the target vehicle and radar antenna.

Target Vehicle Blast

A table of blast radius versus burst height must be provided (fig. 8). If the burst point occurs within the radius specified for the determined height of burst, then a blast kill of the target vehicle is deemed to have occurred for that sample simulation with probability, p (fig. 9 and User Guide, app B).

Radar Blast

A function of the form illustrated in figure 10 must be provided for this option. This function defines radar blast kill probability as a function of range from the antenna to the burst point. For each simulation, radar blast kill is determined from the specified function.

Fragmentation

Fragmentation effects are determined from the results of preliminary MAE analysis of the fragmenting warhead. The MAE computer code is described in references 1 and 2. The MAE program computes conditional kill probabilities as a function of burst point proximity to target center, burst height, attack elevation angle, and projectile terminal velocity. With the MAE code for a given terminal scenario for each of several burst heights, a suitable representation of the fragmentation P_k function can be described. For each burst height, a P_k grid is computed which provides the basis for the construction of a P_k box grid about the target center. It is then a simple matter of interpolating in the range, deflection and height directions as well as for elevation angle to estimate the fragmentation P_k for the actual burst point (fig. 11). Fall-off P_k along the edges of the P_k box is assumed to be linear out to a specified limit; that is, a limit is specified in the range, deflection, and height directions at which the fragmentation P_k drops to zero.

It is important to note that the fragmentation kill probabilities generated by the MAE program are based on vulnerability data averaged over all attack azimuths. Also, P_k's are determined by the MAE code by computation of the proximity

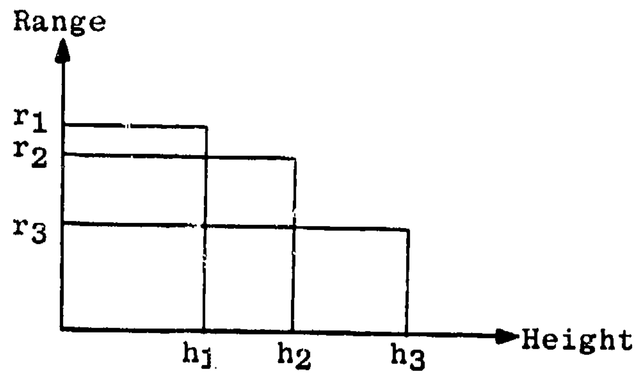

Figure 8. Blast radii vs height

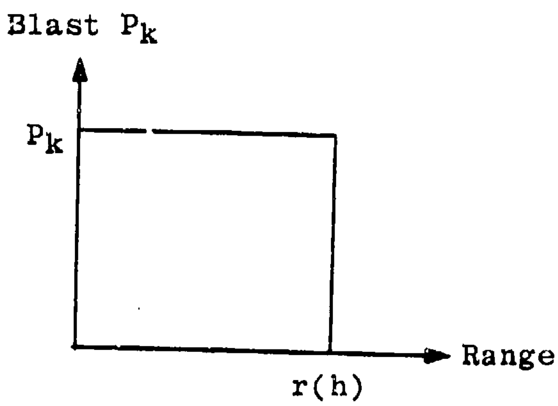

Figure 9. Blast kill probability vs height

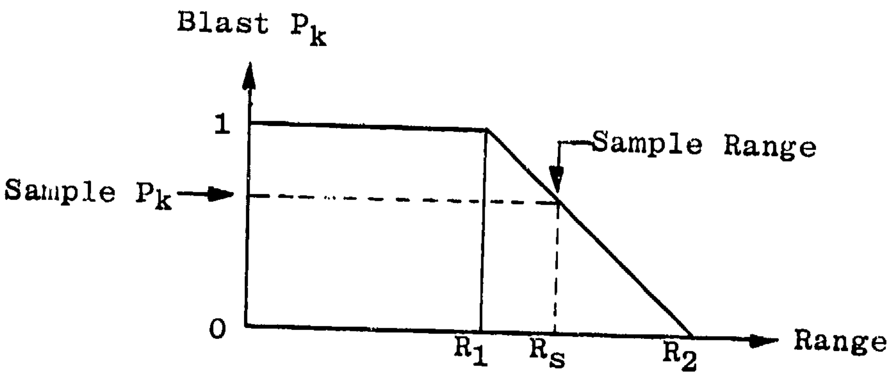

Figure 10. Radar blast function

Estimation of Fragmentation P_k

By Triple Interpolation

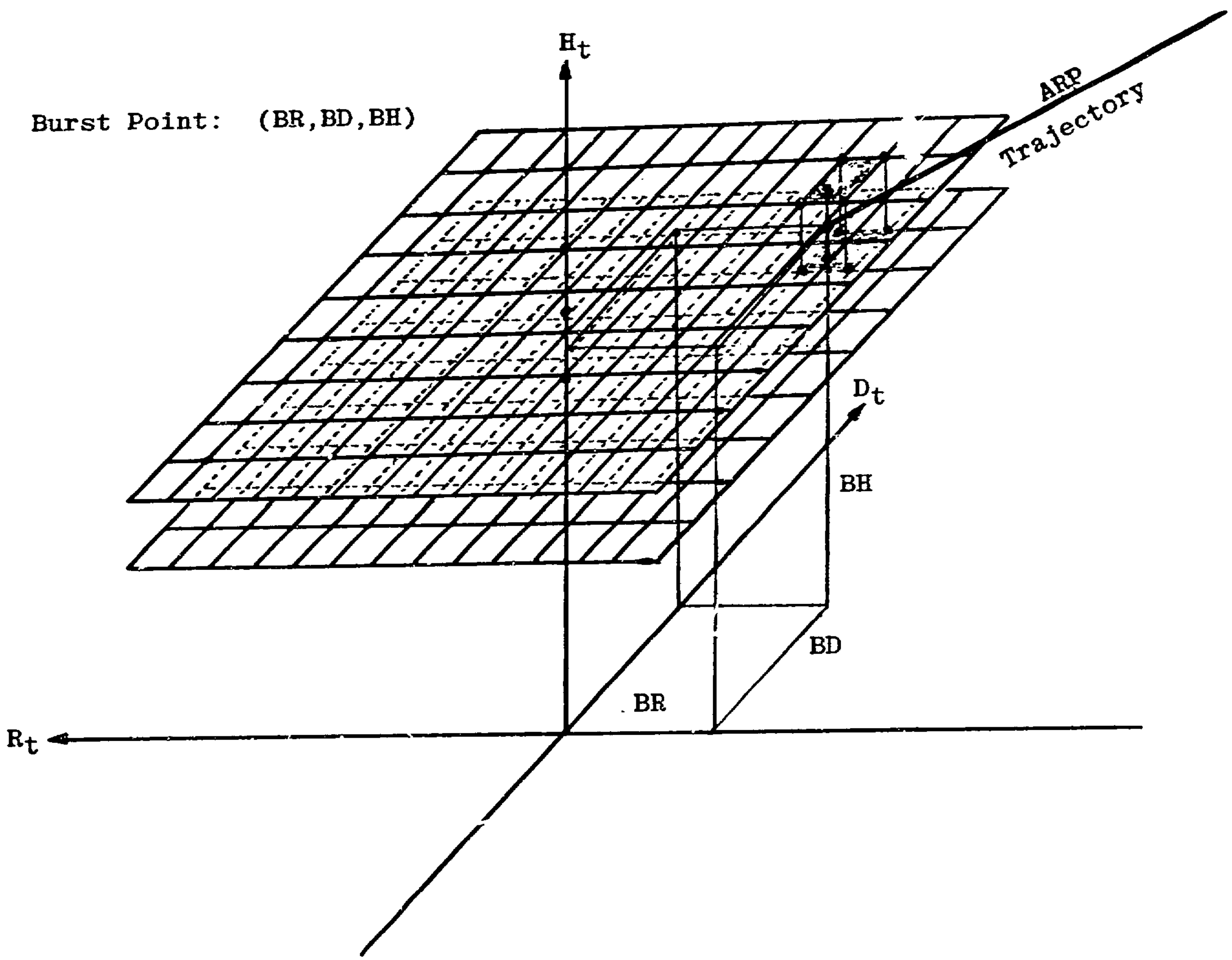

Figure 11. Fragmentation grid interpolation

of the burst point to the target center of vulnerability point. This point-to-point relationship is deficient for narrow spray angle munitions in close proximity to the target. Also, since ARPSIM assumes a particular attack azimuth, the assumption is made that, for the purposes of the ARPSIM model, the average vulnerability of the target can be used to represent the vulnerability for any particular attack azimuth.

As an alternative to the P_k grid box, the MAE program can be used to generate a P_k-versus-range function, where the P_k is averaged over all target azimuths. ARPSIM can utilize this function to interpolate for P_k based upon the range from the burst point to the target center. The P_k-versus-range function can be generated for various burst height and elevation angle combinations. This approach is not recommended with directional warheads.

When the MAE program is used, the blast option available with the MAE code should not be used.

MONTE CARLO ESTIMATES

The program flow procedures are followed for each simulation to provide estimates for direct hit, body blast, radar blast, and fragmentation effects in the form of kill probabilities, P_k. Estimates of these kill probabilities are computed by using

$$P_k(n) = \frac{\sum_{i=1}^{n} P_k(i)}{n}, \quad n = \text{sample size}$$

for each of the kill mechanisms. The combined kill probability is computed for each sample using

$$P_k(i) = 1-[1-P_{DH}(i)]\ [1-P_{RDR}(i)]\ [1-P_{BLST}(i)][1-P_F(i)]$$

These overall kill probabilities are averaged for each individual component kill probability to give Monte Carlo estimates of the effectiveness of the individual kill mechanisms as well as the overall probability of defeating the target.

CONCLUSIONS

The ARPSIM model can be used to provide both weapon designers and effectiveness analysts with an assessment of the potential for the ARP system. As a design tool, ARPSIM provides insight into the contributions of guidance and fuzing policies to the overall performance of the ARP warhead. ARPSIM does not simulate the guidance and control or radiation sensing mechanisms. ARPSIM does provide a means to parametrically assess the relative importance of various performance levels of the guidance, fuzing, and warhead functions. By providing

effectiveness information for a host of performance capabilities, ARPSIM is a useful tool to aid in exploiting those elements of the system which provide the greatest payoff in terms of system effectiveness. ARPSIM can also be utilized to provide data for systems analyses once performance criteria for guidance, fuzing, and warhead functioning have been firmly established by weapon design.

The following specific assumptions and limitations are imbedded within the ARPSIM model:

1. Target is engaged in open flat terrain.

2. ARP terminal trajectory is linear with the longitudinal axis of the projectile collinear with the trajectory.

3. The target configuration can be adequately represented by an aggregate of rectangular parallelepipeds.

4. Fragmentation effects can be estimated with the use of either a P_k box or a P_k-versus-range function generated by the material lethal area program based upon vulnerability data averaged over all azimuths.

RECOMMENDATIONS

The computer code follows a sequence of steps for each sample simulation. Any of these steps can be treated as a separate functional module (fig. 1). The degree of simulation detail can be changed by developing more complex modules to either increase simulation accuracy or expand modular function. The consequences of either improving the model's resolution or expanding its scope are an increase in computer processing time and a resultant increase in the cost of analysis. These consequences must be weighed against the advantages to be gained from the refinement of the model.

Some refinements which might be of merit include the direct computation of fragmentation effects (rather than use the results of precomputations with the MAE code) and the capability to define a complex target array consisting of a multiplicity of target elements.

REFERENCES

1. R. D. Webster, "An Overlay Computer Program for Fragmentation Reduction, Lethal Area, and Target Effects Computations," Information Report E2, Systems Effectiveness Branch, LCWSL, ARRADCOM, Dover, NJ, revised February 1980 by William Matzkowitz.

2. "Computer Program for General Full Spray Materiel MAE Computations, Vol 1, Users Manual," Manual 61 JTCG/ME-79-1-1, Joint Technical Coordinating Group for Munitions Effectiveness, 18 January 1979.

APPENDIX A

USER GUIDE

This user guide is intended to aid those who have access to the ARRADCOM CDC 6500/6600 central computing facility via INTERCOM and BATCH mode processing. Others who may wish to use or modify ARPSIM for operation on a different computer system should also find this guide informative and helpful.

For assessment of fragmentation effects with ARPSIM, it is first necessary to generate files containing fragmentation P_k data as determined by the materiel MAE code (ref 1). There are two alternate forms that the MAE-produced P_k data may take for use by ARPSIM:

1. A P_k grid where grids are defined for the ARP terminal elevation attack angle for up to four different burst heights.

2. A P_k versus range table defined for these same terminal conditions.

For directional fragmentation patterns, the P_k grid format provides a better estimate of the effects produced by the non-symmetry of the warhead effects pattern. P_k functions produced by the MAE code are developed as follows:

P_k Grid Function

Several options exist with the MAE code described in reference 1 which allow the user to define the bounds of the P_k grid in a variety of ways. It is important to note that ARPSIM is limited to a grid size of no more than 20 cells in either range or deflection directions. It is quite possible that fragmentation effects for an ARP warhead might exist at ranges far in excess of the actual miss distance from the target being attacked. For this reason, the user is advised to analyze the guidance errors and fuzing scheme being considered in order to determine practical limits to the size of the P_k grid. Input data for the MAE code are often in units of feet, whereas the P_k grid boundaries which are output are metric. Also, ARPSIM can be used with any consistent set of units, although it is recommended that the metric system be used. It is advisable then to predetermine the practical range for a P_k grid and then use an option with the MAE code to define the limits of the P_k grid.

When using the OVRLAY code described in reference 1 to make MAE calculations, the MTRX option should be called for but not actually used; that is, the MTRX input data should consist of a blank card. For the user who is not familiar with the OVRLAY system of computer codes, it is a system that was established to provide users with the capability to make single computer runs beginning with raw fragmentation data continuing through MAE computations and the development of P_k grids, and culminating with estimates of artillery system effectiveness against certain target arrays. The overlay technique is used to combine a number of computer codes devoted to these analyses. The MTRX option signals the MAE code to produce a P_k grid on a file named TAPE4 in formats which are compatible with both the MTRX and ARPSIM codes. For this reason, the user should call for the MTRX option when using the OVRLAY code, and then provide only a blank card as the input for the MTRX code. By doing this, the user will normally terminate the OVRLAY code and will have defined a TAPE4 file consisting of a string of P_k grids, one for each burst height. It is advisable to save the TAPE4 file as a permanent file for future recall of the data, as necessary, when using the ARPSIM code.

If P_k grids are being generated for several (up to three) different attack elevations for use by ARPSIM, each elevation angle data set should be generated by a separate MAE run. Then, when recalling the P_k grid files, define the data on file TAPE2 for the lowest angle data, TAPE3 for the next lowest, and TAPE4 for the highest. Burst heights should always be computed in the order of lowest to highest.

For users who do not have access to the MAE code or who will use an alternate code to generate P_k grids, the files TAPE2, TAPE3, and TAPE4 should contain, sequentially, a card image data record in format (2I3) indicating the number of grid coordinates in range and deflection.

Next, are two card sets in format (10F7.1) where the first set defines the range coordinates of the grid boundaries and the second set defines the deflection boundaries. Boundaries are defined from lowest to highest values. Following these data sets are the P_k's associated with the grid in format (10F7.5) where P_k's are given first for the first range cell (lowest grid bracket) for each of the deflection cells (again, beginning with the lowest bracket) and proceeding through all range brackets in the same manner. All P_k grids are defined this way for each burst height in order of lowest to highest burst height.

P_k Versus Range

An average P_k versus range function (table) can be used if the number of ranges is no greater than 200. Format for data entry is (F8.3, F8.5) where the first item is range (usually in meters) followed by the corresponding average P_k. The MAE code can generate this table on a file named TAPE15. These files can be saved, like the grid files, and recalled when using the ARPSIM. These files when used other than with the MAE code or when recalling MAE-generated files, are defined like the grid files, i.e., lowest angle data on TAPE2, next lowest on TAPE3, and highest on TAPE4. Each burst height (up to four) has its own table defined beginning with the lowest burst height and stored sequentially on each file.

Following definition of the P_k functions on TAPE2, TAPE3, and TAPE4 (as required), the ARPSIM can be exercised using a teletype (TTY). Preliminary steps required to run ARPSIM on the ARRADCOM computer in INTERCOM mode are as follows:

INTERCOM Mode Setup

The following sequence is required to access the ARPSIM code and begin its execution:

```
LOGIN.
...follow normal login procedures
COMMAND - ETL,500.
COMMAND - FETCH,ARP,BWEBSTER.
COMMAND - ATTACH,T,TAPE1FILE,ID=your id.
COMMAND - COPYBF,T,TAPE1.
COMMAND - RETURN,T.
```

```
COMMAND - ATTACH,TAPE2,...
COMMAND - ATTACH,TAPE3,...
COMMAND - ATTACH,TAPE4,...
COMMAND - ARP
```

The sequence from ATTACH,T... through RETURN,T. is only required if a previously defined set of basic inputs is to be used as a basis for this run. Also, the sequence ATTACH,TAPE2,... through ATTACH,TAPE4,... is required only in accordance with the requirements to estimate fragmentation effects and the diversity of attack elevations required.

In response to the command ARP, the user will be given the opportunity to produce a summary input guide. Following that, the user will be asked whether a file named TAPE1 is to be used as the basis for input data. This option is provided as an aid to the user who expects to make several computer runs with the model using the same basic input data set. The ARPSIM code has a built-in input editing routine which continually redefines the file TAPE1 to be the current basic input data set. The user who wishes to make additional runs with a basic data set merely has to define the current data set and then, after ARPSIM has been run, the TAPE1 file is stored on a permanent file for later use as with the ATTACH,T... through RETURN,T. sequence described above. If a basic data set is being used, then the initial input conditions are listed. Then, in all cases, the user is asked to ENTER DATA OR END - . In response to this command the user begins to enter "word" type data to either initialize a data type or change a data type. Word type data which can be entered are defined according to general function in the section which follows. Formats are (A4,F10.4).

"Word" Type Data

This section is divided into functional areas as follows:

Guidance Data

NGER,n. NGER signifies the number of guidance error data sets to be input. The value of n equals the number of different guidance error sets to be analyzed.

NCEP,1. If guidance errors are input as standard deviations in both deflection and height, omit this set. If errors are input as CEP, then include this set. Note that in all cases errors are defined in a plane passing through a homing point and normal to the ARP flight path.

Fuzing Scheme

FZAM,n. FZAM signifies the use of the fuzing angle primary fuze where n is the mean value of the fuze half-vertex angle; i.e., n is

the mean angle from the ARP trajectory to the fuzing glitter point at which fuzing will occur. Units are degrees.

FZAS,n. FZAS signifies the standard deviation of fuzing angle associated with the mean value defined by FZAM, where the value n is the standard deviation. Units are degrees.

FZTM,n. FZTM signifies the use of a linear (or time) fuze where the sign of the value of n indicates whether the fuze operates in the vertical direction or along the trajectory. A negative n signifies the vertical option. The value of n is the mean distance from the guidance plane (or initial fuzing point if used in conjunction with the FZAM option) in the negative range direction where fuzing occurs. With the vertical option, the distance is measured from the ground. A time fuze operating along the ARP trajectory can be simulated by converting the values to distances by using the known ARP terminal velocity.

FZTS,n. FZTS defines the standard deviation associated with the FZTM data in all modes.

PKPF,n. The value of n is the probability that the primary fuze (options described above) will function.

PDVT,n. Selects the backup fuze option. The value for n is 0 for a PD (ground burst) backup and is the number of entries in a height versus probability table (up to 5 values) to define the VT fuze functioning distribution.

GLTR,n. Specifies the glitter points used by the angular fuzing function option. If n is 0, the fuze functions relative to the point (0,0,TGTC) where TGTC is the center of target vulnerability. If n is non-zero, the fuze functions relative to one of the n input glitter points. A positive n signifies that the fuzing glitter point is selected randomly; a negative n signifies that the first glitter point encountered will cause fuzing.

Terminal Conditions

OMEG,n. The elevation angle measured from the ground is chosen from a normal distribution with mean value n.

OMGS,n. The standard deviation associated with OMEG is input as n.

TGTC,n. The center of target vulnerability is input as a height above the origin at (0,0,n). If direct hit effects are not being analyzed (direct hit boxes are not defined), then the vehicle blast effects are determined based on the range from the burst point to (0,0,TGTC).

DHAZ,n. The azimuth angle-of-attack is n and is measured from the negative range axis in the direction of the positive deflection axis. Units are degrees. To choose the azimuth uniformly random between 0 and 360 degrees, set n = -1.

DUDR,n. The dud rate of ARP projectiles is given as n, where a 5% dud rate corresponds to n = 0.05.

General Conditions

SAMP,n. The number of Monte Carlo samples is n.

PRNT,1. Specifies that only a final summary of results is to be output.

SRNG,n. Tables of average combined P_k can be output as a function of azimuth, elevation and range as well as averaged over non-zero results obtained in the angular bins for each range. The value for n is the upper limit (defaults to 100) for range information. The range scale is logarithmic and includes 10 bins, beginning with the minimum range obtainable (considering direct hit implications) and ending at n.

Fragmentation Effects

PKNH,n. Specifies the number of heights, n, at which fragmentation effects are provided (either as P_k grids or P_k versus range tables). Must not exceed 4.

PKNA,n. Specifies the number of elevations, n, for which fragmentation effects are provided. Must not exceed 3.
For n = 1, effects are on TAPE2.
For n = 2, effects are on TAPE2 for lowest angle data and on TAPE3 for highest angle data.
For n = 3, effects are on TAPE2 for lowest angle data, TAPE3 for middle angle data, and TAPE4 for highest angle data.

FUNC,1. Selects option to use P_k versus range tables for fragmentation effects in place of the P_k grids.

Direct Hit Effects

DHIT,n. Specifies the number of target boxes to be input to approximate the shape of the target for purposes of computing direct hit effects. Boxes are defined relative to (0,0,0) and the total number of boxes cannot exceed 5.

PKDH,n. Direct hit P_k if a direct hit is achieved. If n = 0, P_k is defaulted to one.

Blast Effects

PKBL,n. Specifies the blast P_k if the burst point is within a range specified by the BLST data of the surface of any direct hit box. If direct hit boxes are not used, then range is calculated to the point (0,0,TGTC).

BLST,n. Specifies the range from the direct hit surfaces or the point (0,0,TGTC) within which the blast P_k against the vehicle body is that given by the PKBL data. To enter a table of blast ranges versus burst height, enter a negative value for n which corresponds to the number of entries in the blast range versus height table (may not exceed 5).

RADR,1. Include to compute blast effects against radar antenna separately from vehicle blast.

End of Word Data

END Must always be included at the end of the "word"-type data entries.

After all "word"-type data have been entered, the code will ask for certain data which are required by some of the options chosen by the "word" cards. These additional input requirements are discussed in the following section. All data are free-formatted.

Guidance Data

Either pairs of deflection and height standard deviations are entered or, if NCEP,1. data is entered in the "word" section, then the guidance errors are input as CEP's.

The homing point coordinates follow the guidance error inputs. The homing point is generally the coordinates of the center of the radar antenna.

Direct Hit Boxes

The limits of the dimensions of each direct hit box are input for range, deflection, and height, respectively. For example, for a direct hit box centered at the origin and having a length of 20 meters, a width of 10 meters, and a height of 5 meters, this data would be input as -10,10,-5,5,0,5.

Radar Data

Radar antenna coordinates are entered for the purposes of radar blast P_k computation.

Following the entry of the radar coordinates, values are entered for two ranges, R1 and R2, which define the radar blast P_k function as being one out to R1 and declining linearly to zero at R2.

Fragmentation

Heights are entered beginning with the lowest value and corresponding to the burst heights used for the MAE computations. An additional height is input last and corresponds to that height at which all fragmentation P_k's are zero.

Following the height data, two values are input corresponding to the distances beyond the edge of the P_k grids where the fragmentation P_k becomes zero in range and deflection, respectively.

Elevation angles are entered next, beginning with the lowest angle and corresponding to the angles for which the MAE code was run to produce the fragmentation P_k data.

VT Backup Fuzing

A table of probability of fuze functioning at height less than or equal to height, H, is used to generate VT fuzing data. Up to five heights are input followed by probabilities corresponding to the probability of fuze functioning between the respective height and the next lower height. Ideally, probability values should sum to unity.

Glitter Points

Glitter point coordinates are entered for each glitter point. All coordinates are relative to (0,0,0) of the target.

Blast Data (Vehicle)

If the blast-distance-versus-burst-height option is chosen (negative n on BLST,n data), then n pairs of blast distance, height are entered.

This concludes the input requirements for using the ARPSIM model. Word type data can be changed or input in any order. Required additional data will be

prompted from the user by the code. The user is always given the option of listing the current data set (with the exception of the fragmentation P_k data) or changing the data set prior to actual computations. When the computations are completed for all cases, the user is given the opportunity to run additional cases based on the current data sets.

APPENDIX B

EXAMPLE

The following example, provided as a supplement to the User Guide in Appendix A, denotes the type of material generated for a typical ARPSIM run:

```
**************************************************
*     ANTI-RADIATION SIMULATION PROGRAM - 9/1/80*

**************************************************
*    NOTE:  ALL COORDINATES ARE DEFINED RELATIVE TO*
*           ORIGIN AT GROUND ZERO OF TARGET.*
*           COORDINATE SYSTEM IS RECTANGULAR,*
*           TARGET HEADING IS NEGATIVE RANGE,*
*           DRIVER SIDE (L) IS POSITIVE DEFLECTION.*
*           HEIGHT IS MEASURED FROM GROUND.*

**************************************************

        DATE -  08/27/80
        TIME -  13.47.13.

**************************************************
*DO YOU WANT A LISTING OF CODE NAMES? *Y

**************************************************
*OMEG - MEAN ATTACK ANGLE*
*OMGS - ATTACK ANGLE STD DEV*
*  NOTE:  FOLLOWING GUIDANCE ERROR PARAMETERS*
*          (SIGD,SIGH) ARE MEASURED*
*          IN PLANE NORMAL TO TRAJECTORY AND*
*          PASSING THROUGH HOMING POINT*
*NGER - NUMBER OF GUIDANCE ERRORS TO CONSIDER*
*         ENTER HOMING POINT (R,D,H), GUIDANCE*
*         ERRORS ARE DISTRIBUTED ABOUT HOMING PT.*
*NCEP - 1., IF CEP IS INPUT FOR GUIDANCE ERROR SIGMAS*
*FUNC - 1., IF OPTION TO USE PK VS. RANGE DAMAGE*
*            IN PLACE OF PK BOX FUNCTION IS SELECTED*
*            YOU MUST DEFINE PK VS R DATA FOR EACH*
*            HEIGHT LAYER SPECIFIED BY PKNH CARD*
*            AND EACH ANGLE SPECIFIED BY*
*            PKNA CARD.*
*FZAM,FZAS,FZTM,FZTS - FUZING ERROR OPTIONS*
*  FZAM - MEAN ANGLE AT WHICH FUZING OCCURS ON*
*           INTERCEPT*
*  FZAS - STD DEV ASSOCIATED WITH FZAM*
*NOTE:  FOR UNIFORM FUZING ANGLE BETWEEN FZAM*
*       AND FZAS, ENTER A NEGATIVE VALUE FOR FZAM*
*       FUZE ANGLE WILL BE CHOSEN UNIFORMLY RANDOM*
*       BETWEEN POSITIVE FZAM AND FZAS*
*NOTE:  FUZING PLANE PASSES THROUGH FUZING GLITTER*
*       POINT NORMAL TO SAMPLE TRAJECTORY*
*  FZTM - MEAN DISTANCE FROM GUIDANCE PLANE AT WHICH*
*          FUZING WILL OCCUR ALONG TRAJECTORY*
*NOTE:  ENTER A NEGATIVE FZTM FOR HEIGHT FUZING *
*       WITH MEAN HEIGHT ABS(FZTM) *
```

```
*  FZTS - STD DEV ASSOCIATED WITH FZTM*
*SAMP - SAMPLE SIZE*
*PKNH - NUMBER OF HEIGHTS AT WHICH FRAGMENTATION*
*       PK DATA WILL BE DEFINED*
*       NOTE: PKNH < 5*
*PKNA - NUMBER OF ELEVATION ANGLES FRAGMENTATION*
*       PK DATA WILL BE DEFINED FOR*
*       NOTE: PKNA < 4*
*PKPF - PROBABILITY OF PRIMARY FUZE FUNCTIONING*
*PDUT - 0. FOR PD BACKUP, NVT FOR VT BACKUP FUZE*
*       WHERE NVT = NUMBER OF VT BURST HEIGHTS*
*GLTR - 0. IF PRIMARY FUZE FUNCTIONS RELATIVE TO*
*       CENTER OF TARGET, NGLT IF PRIMARY FUZE*
*       FUNCTIONS  RELATIVE TO ANY ONE OF NGLT*
*       EQUALLY LIKELY GLITTER POINTS .*
*       SET NGLT NEGATIVE TO PICK FIRST*
*       POINT ENCOUNTERED.*
*SRNG - MAXIMUM RANGE FOR COMPUTING PK VS RANGE*
*PRNT - 1. TO PRINT SUMMARY ONLY, 0. OTHERWISE*
*DBUG - 6. TO PRINTOUT PROGRAM DEBUGGING DATA*
*       DBUG = 1, GUIDANCE & FUZING DATA*
*       DBUG = 2, DIRECT HIT PENETRATION DATA*
*       DBUG = 3, HOMING ANGLE DATA*
*       DBUG = 4, PK BOX DATA*
*       DBUG = 5, PK GRIDS*
*       DBUG = 6, PK VS R DATA*

*TGTC - HEIGHT OF TARGET CENTER ABOVE GROUND*
*DUDR - DUD RATE OF PROJECTILE, EXPRESSED AS A FRACTION*
*DHIT - DIRECT HIT OPTION, NUMBER OF TARGET BOXES*
*       IF DHIT IS OMITTED AND BLST IS INCLUDED,*
*       BLST IS RADIUS FROM (0,0,TGTC) WITHIN*
*       WHICH PKBLST = 1.*
*PKDH - DIRECT HIT PK (0. - 1.)*
*PKBL - BLAST PK (0. - 1.)*
*RADR - 1., DEFINE FUNC FOR BLAST KILL OF RADAR ONLY*
*       AND READ IN RADAR ANTENNA COORDINATES.*
*       TO DEFINE FUNC, SPECIFY R1 AND R2,*
*       WHERE BLAST PK IS 1 OUT TO R1 AND*
*       DECLINES LINEARLY TO 0 AT R2.*
*DHAZ - AZIMUTH ANGLE OF ATTACK OFF FRONT OF TARGET*
*       TOWARD DRIVER SIDE.  SET TO -1. FOR RANDOM*
*BLST - BLAST RADIUS WITHIN WHICH VEHICLE PK=PKBL*
*NOTE:  TO ENTER BLAST RADII VS. BURST HEIGHT,*
*       ENTER NEGATIVE NUMBER OF BLAST,HGT PAIRS*
*       IN PLACE OF VALUE OF BLST.  PAIRS OF*
*       BLAST,HGT ARE ENTERED IN ASCENDING ORDER*
*       OF HEIGHT.*
*COORDINATE SYSTEM IS RECTANGULAR, *
*TARGET HEADING IS NEGATIVE RANGE, *
*DRIVER SIDE (LEFT) IS POSITIVE DEFLECTION,*
*HEIGHT IS MEASURED FROM GROUND*
*ENTER DATA BY ENTERING CODE NAME*
*FOLLOWED BY A COMMA AND THE VALUE IN FLOATING*
*POINT FORMAT.  TO END DATA ENTRY, ENTER *
*THE WORD END IN COLUMNS 1-3*
*DO YOU WISH TO INITIALIZE DATA FROM SAVED*
*DATA FILE (TAPE1)? *Y


 *INITIAL INPUTS - *
 FZAM      70.000
 PKDH       1.000
 PKBL       1.000
 FZAS      10.000
 OMEG      10.000
 NGER       3.000
 NCEP       1.000
 FUNC       1.000
 DHIT       2.000
 SAMP     100.000
 PKNH       4.000
 PKNA       3.000
 PDUT       5.000
 PKPF        .950
 GLTR       3.000
 SRNG     100.000
 TGTC      10.000
 DUDR        .050
 BLST       3.000
 END
```

```
3. 6. 9.
0. 0. 10.
-5. 5. -5. 5. 0. 10. -2. 2. -2. 2. 10. 20.
4. 8. 12. 16.
0. 10. 20.
2. 4. 6. 8. 10.
.2 .2 .2 .2 .2
-5. -5. 10. -5. 5. 0. -2. 2. 20.
*DO YOU WANT TO CHANGE ANY DATA? - *Y

*ENTER DATA OR END - *RBDR,1.

*ENTER DATA OR END - *END

RADA DATA -
ENTER RADAR ANTENNA COORDINATES (R,D,H) RELATIVE
TO TARGET GROUND ZERO. -0.,0.,20.

ENTER R1,R2, WHERE RADAR BLAST PK=1
OUT TO R1 AND DECLINES LINEARLY
TO ZERO AT R2 -10.,20.

*DO YOU WANT CURRENT DATA LISTED? *N

*DO YOU WANT TO CHANGE ANY DATA? - *N


**************************************************

*FINAL RESULTS*

PK =  .7810  PKSD =  .0350  NSAMP =     100



**************************************************

*DO YOU WANT PK VS R, ALPHA, BETA? *Y



*  PK        R          ALPHA              BETA*
*------    -----    ---------------    ---------------*
1.0000     11.8      60.0 -   90.0     60.0 -   75.0
1.0000     11.8     120.0 -  150.0     15.0 -   30.0
1.0000     11.0     120.0 -  150.0     30.0 -   45.0
1.0000     11.0     120.0 -  150.0     45.0 -   60.0
1.0000     11.8     150.0 -  182.0     45.0 -   60.0
1.0000     11.8     150.0 -  192.0     60.0 -   75.0
1.0000     11.8     180.0 -  210.0     60.0 -   75.0
1.0000     11.8     210.0 -  240.0     30.0 -   45.0
1.0020     11.8     210.0 -  240.0     45.0 -   60.0
 .7435     12.7     120.0 -  150.0     30.0 -   45.0
1.0000     12.7     120.0 -  150.0     45.0 -   60.0
 .9450     12.7     150.0 -  180.0     45.0 -   60.0
1.0000     12.7     150.0 -  180.0     60.0 -   75.0
1.0000     12.7     210.0 -  240.0     45.0 -   60.0
1.0000     12.7     210.0 -  240.0     75.0 -   90.0
1.0000     14.1     120.0 -  150.0     45.0 -   60.0
 .9051     14.1     150.0 -  180.0     45.0 -   60.0
1.0000     14.1     150.0 -  180.0     60.0 -   75.0
1.0000     14.1     210.0 -  240.0     45.0 -   60.0
 .9814     16.2     120.0 -  150.0     45.0 -   60.0
 .7058     10.2     150.0 -  180.0     30.0 -   45.0
 .8399     16.2     150.0 -  190.0     45.0 -   60.0
1.0000     16.2     150.0 -  180.0     60.0 -   75.0
 .6365     16.2     180.0 -  210.0     30.0 -   45.0
1.0000     16.2     210.0 -  240.0     45.0 -   62.0
1.0000     16.2     210.0 -  240.0     60.0 -   75.0
 .8806     19.7     120.0 -  150.0     45.0 -   60.0
1.0000     19.7     120.0 -  150.0     60.0 -   75.0
 .5645     19.7     150.0 -  180.0     30.0 -   45.0
 .7680     19.7     150.0 -  180.0     45.0 -   60.0
 .4257     19.7     180.0 -  210.0     30.0 -   45.0
 .9828     19.7     210.0 -  240.0     45.0 -   60.0
1.0000     19.7     210.0 -  240.0     60.0 -   75.0
 .2159     25.1     150.0 -  180.0     15.0 -   30.0
 .1064     33.5     150.0 -  180.0     15.0 -   30.0
 .1227     33.5     180.0 -  210.0     15.0 -   30.0
 .0315     46.7       0.0 -   30.0      0.0 -   15.0
```

****

'AVG PK VS. R'

1.0000	11.8
.9676	12.7
.9603	14.1
.9030	16.2
.7332	19.7
.2159	25.1
.1104	33.5
.0315	46.7

****

'FINAL RESULTS'

PK = .6970 PKSD = .6409 NSAMP = 100

****

'DO YOU WANT PK VS R, ALPHA, BETA? 'Y

'PK	R	ALPHA	BETA'
1.0000	11.8	120.0 - 150.0	15.0 - 30.0
1.0000	11.8	120.0 - 150.0	30.0 - 45.0
1.0000	11.8	120.0 - 150.0	45.0 - 60.0
1.0000	11.8	120.0 - 150.0	60.0 - 75.0
1.0000	11.8	150.0 - 180.0	30.0 - 45.0
1.0000	11.8	150.0 - 180.0	45.0 - 60.0
1.0000	11.8	150.0 - 180.0	60.0 - 75.0
1.0000	11.8	210.0 - 240.0	30.0 - 45.0
1.0000	12.7	120.0 - 150.0	30.0 - 45.0
.6573	12.7	150.0 - 180.0	15.0 - 30.0
1.0000	12.7	150.0 - 180.0	60.0 - 75.0
1.0000	12.7	210.0 - 240.0	45.0 - 60.0
1.0000	14.1	90.0 - 120.0	60.0 - 75.0
1.0000	14.1	120.0 - 150.0	45.0 - 60.0
1.0000	14.1	120.0 - 150.0	60.0 - 75.0
.8377	14.1	150.0 - 180.0	45.0 - 60.0
1.0000	14.1	150.0 - 180.0	60.0 - 75.0
1.0000	14.1	210.0 - 240.0	45.0 - 60.0
1.0000	16.2	90.0 - 120.0	60.0 - 75.0
.9349	16.2	120.0 - 150.0	45.0 - 60.0
.6972	16.2	150.0 - 180.0	30.0 - 45.0
.8543	16.2	150.0 - 180.0	45.0 - 60.0
1.0000	16.2	150.0 - 180.0	60.0 - 75.0
.4121	16.2	180.0 - 210.0	15.0 - 30.0
1.0000	16.2	210.0 - 240.0	45.0 - 60.0
1.0000	16.2	210.0 - 240.0	60.0 - 75.0
.9039	19.7	120.0 - 150.0	45.0 - 60.0
1.0000	19.7	120.0 - 150.0	60.0 - 75.0
1.0000	19.7	120.0 - 150.0	75.0 - 90.0
.3305	19.7	150.0 - 180.0	15.0 - 30.0
.3953	19.7	150.0 - 180.0	30.0 - 45.0
.8526	19.7	150.0 - 180.0	45.0 - 60.0
1.0000	19.7	150.0 - 180.0	60.0 - 75.0
1.0000	19.7	150.0 - 180.0	75.0 - 90.0
.2522	19.7	180.0 - 210.0	0.0 - 15.0
.9565	19.7	210.0 - 240.0	45.0 - 60.0
1.0000	19.7	210.0 - 240.0	60.0 - 75.0
1.0000	25.1	120.0 - 150.0	60.0 - 75.0
1.0000	25.1	120.0 - 150.0	75.0 - 90.0
.2430	25.1	150.0 - 180.0	15.0 - 30.0
.9414	25.1	150.0 - 180.0	45.0 - 60.0
.2338	25.1	180.0 - 210.0	0.0 - 15.0
.1777	25.1	180.0 - 210.0	15.0 - 30.0
.2979	25.1	180.0 - 210.0	30.0 - 45.0
1.0000	25.1	210.0 - 240.0	60.0 - 75.0
.1079	33.5	150.0 - 180.0	15.0 - 30.0
.0794	33.5	150.0 - 180.0	45.0 - 60.0
.1050	33.5	180.0 - 210.0	0.0 - 15.0
.1427	33.5	180.0 - 210.0	15.0 - 30.0
.9911	33.5	210.0 - 240.0	60.0 - 75.0
.0361	46.7	0.0 - 30.0	0.0 - 15.0
.0339	46.7	150.0 - 180.0	0.0 - 15.0
.0232	46.7	180.0 - 210.0	0.0 - 15.0
.0073	67.5	0.0 - 30.0	0.0 - 15.0
.0176	67.5	150.0 - 180.0	0.0 - 15.0
.0149	67.5	180.0 - 210.0	0.0 - 15.0
.0015	100.0	150.0 - 180.0	0.0 - 15.0

$$

'AVG PK VS. R'

1.0000	11.8
.9429	12.7
.9797	14.1
.9215	16.2
.8335	19.7
.6480	25.1
.2342	33.5
.0365	46.7
.0132	67.5
.0015	100.0

$$

'FINAL RESULTS'

PK = .6983 PKSD = .0397 NSAMP = 100

$$

'DO YOU WANT PK VS R, ALPHA, BETA? 'Y

' PK	R	ALPHA	BETA'
1.0000	11.8	120.0 - 150.0	15.0 - 30.0
1.0000	11.8	120.0 - 150.0	30.0 - 45.0
1.0000	11.8	120.0 - 150.0	45.0 - 60.0
1.0000	11.8	150.0 - 180.0	45.0 - 60.0
.6204	11.8	210.0 - 240.0	15.0 - 30.0
1.0000	11.8	210.0 - 240.0	30.0 - 45.0
1.0000	12.7	120.0 - 150.0	30.0 - 45.0
1.0000	12.7	120.0 - 150.0	45.0 - 60.0
.5632	12.7	150.0 - 180.0	0.0 - 15.0
.7695	12.7	150.0 - 180.0	30.0 - 45.0
.8211	12.7	150.0 - 180.0	45.0 - 60.0
1.0000	12.7	210.0 - 240.0	45.0 - 60.0
.4675	14.1	30.0 - 60.0	15.0 - 30.0
1.0000	14.1	120.0 - 150.0	45.0 - 60.0
.6169	14.1	150.0 - 180.0	15.0 - 30.0
1.0000	14.1	210.0 - 240.0	45.0 - 60.0
.9818	16.2	120.0 - 150.0	45.0 - 60.0
1.0000	16.2	120.0 - 150.0	60.0 - 75.0
.4140	16.2	150.0 - 180.0	15.0 - 30.0
.6730	16.2	150.0 - 180.0	30.0 - 45.0
.8445	16.2	150.0 - 180.0	45.0 - 60.0
1.0000	16.2	150.0 - 180.0	60.0 - 75.0
1.0000	16.2	210.0 - 240.0	45.0 - 60.0
1.0000	16.2	210.0 - 240.0	60.0 - 75.0
.9439	19.7	120.0 - 150.0	45.0 - 60.0
1.0000	19.7	120.0 - 150.0	60.0 - 75.0
1.0000	19.7	120.0 - 150.0	75.0 - 90.0
.4614	19.7	150.0 - 180.0	15.0 - 30.0
.3359	19.7	150.0 - 180.0	30.0 - 45.0
.9249	19.7	150.0 - 180.0	45.0 - 60.0
1.0000	19.7	150.0 - 180.0	60.0 - 75.0
1.0000	19.7	150.0 - 180.0	75.0 - 90.0
1.0000	19.7	210.0 - 240.0	60.0 - 75.0
1.0000	25.1	120.0 - 150.0	60.0 - 75.0
1.0000	25.1	120.0 - 150.0	75.0 - 90.0
.2500	25.1	150.0 - 180.0	15.0 - 30.0
.9088	25.1	150.0 - 180.0	60.0 - 75.0
.2247	25.1	180.0 - 210.0	0.0 - 15.0
.1878	25.1	180.0 - 210.0	15.0 - 30.0
.9106	25.1	180.0 - 210.0	60.0 - 75.0
1.0000	25.1	210.0 - 240.0	60.0 - 75.0
.9755	33.5	120.0 - 150.0	60.0 - 75.0
.1173	33.5	150.0 - 180.0	0.0 - 15.0
.1316	33.5	150.0 - 180.0	15.0 - 30.0
.2508	33.5	150.0 - 180.0	45.0 - 60.0
.5454	33.5	150.0 - 180.0	60.0 - 75.0
.5345	33.5	180.0 - 210.0	60.0 - 75.0
1.0000	33.5	210.0 - 240.0	60.0 - 75.0
.0499	46.7	150.0 - 180.0	0.0 - 15.0
.0450	46.7	180.0 - 210.0	0.0 - 15.0
.0389	67.5	150.0 - 180.0	0.0 - 15.0
.0029	67.5	180.0 - 210.0	0.0 - 15.0
.0009	100.0	150.0 - 180.0	0.0 - 15.0

```
************************************************

'AVG PK VS. R'
'.-----------'
 .9578      11.8
 .9592      12.7
 .9169      14.1
 .9130      16.2
 .9061      19.7
 .8519      25.1
 .4844      33.5
 .0482      46.7
 .0009      67.5
 .0009     100.0


************************************************


    RESULTS FOR FOLLOWING CONDITIONS -

           ITEM                MEAN     STD DEV

         ELEVATION          10.0000     0.0000
         FUZE ANGLE         70.0000    10.0000
         LINEAR FUZE         0.0002     0.0000
         AZIMUTH             0.0002
         SAMPLE SIZE  -     100

    HOMING POINT COORDINATES (R,D,H) =     0.0,     0.0,    10.0

    ERROR DATA                      PK    PKFRAG  PKRADR  PKDHIT  PKBLST
    CEP  -   3.0                  .7816   .4173   .0003   .0700   .5400
    CEP  -   6.0                  .6570   .3299   .5517   .0500   .4000
    CEP  -   9.0                  .6583   .2613   .5725   .2100   .3200


************************************************

'DO YOU WISH TO RUN ANOTHER CASE? 'N
```

A description of the material produced by this particular ARPSIM run follows:

Header information is printed, including the time and date of the run. The user is asked whether a listing of input code names is desired (as an aid to generating a proper set of inputs). In this example, the code names are printed. Next, the user is given the option of starting with a previously developed set of inputs which can be changed by a built-in input editing routine. That option is invoked for this example. Note that a file named TAPE1 must be defined which contains this data prior to running ARPSIM. A listing of initial data conditions is provided next. The user is then asked whether any data changes are required.

In this example the user desires to add the capability to estimate radar blast effects. Note that only changed data need be entered at this point. The code then asks for additional information required by the added data. Having fulfilled the data requirements, the user is given the option of listing the entire data set again. Following this, the user is given the option of making any additional changes or corrections to the data set. In this example no additional changes are requested.

Before proceeding with the discussion of the ARPSIM results for this case, a brief run-through is given of the input data set. The FZAM data specifies a fuze angle option with a mean value of 70 degrees for the fuze angle. The FZAS code specifies a 10-degree standard deviation for the fuze angle from simulation to simulation. The PKDH and PKBL data indicate direct hit and vehicle blast P_k's, respectively. Attack elevation of 10 degrees is specified by the OMEG card. NGER indicates three different sets of guidance errors will be analyzed, and NCEP indicates that guidance errors will be input as CEP. FUNC specifies that the fragmentation P_k's will be estimated from interpolations in a set of P_k versus range tables generated by the MAE code for a combination of burst height and elevation angles.

Up to three elevation angle sets can be provided on files TAPE2, TAPE3, and TAPE4. If only a single elevation angle data set is provided, then only TAPE2 is required. Two elevation angles require both TAPE2 and TAPE3. Each file contains P_k versus range for identical burst heights, beginning with the lowest burst height. That is, if four burst heights have been analyzed by the MAE code (the maximum allowable by ARPSIM), each file will contain four P_k versus range tables, one for each burst height beginning with the lowest height and progressing to the highest.

In this example, four burst heights were considered for each of three angles of fall (elevation angles) as specified by the PKNH and PKNA codes, respectively. SAMP provides the number of simulations to run for each case. PDVT specifies that a VT backup fuze is being considered where the height of burst distribution for the backup fuze will be typified at five burst heights. PKPF specifies that the probability that the primary fuze functions is 0.95. GLTR specifies that three glitter points for primary fuzing exist. SRNG gives the maximum range for a P_k versus range table to be generated based upon the results of the ARPSIM run. TGTC provides that the center of target vulnerability is located at 10 (in this case meters) above the target origin (0,0,0). DUDR specifies a projectile dud rate of 5%. BLST provides a blast radius from the TGTC point within which the P_k for vehicle blast effect is as stated on the PKBL data above.

The END code signifies the end of the word type data. The numbers 3., 6., and 9. specify the guidance error CEP's. Following this are the homing point coordinates (0,0,10), and the limits in range, deflection and height of the two direct hit target description boxes. Burst heights and angles of fall (elevations) utilized by the MAE code in generating the P_k versus range tables are specified next. Then the heights and probabilities associated with the backup fuzing function are listed. Finally, glitter point coordinates are specified.

Final results are given as the combined kill probability, the standard deviation of kill probability and the sample size upon which these numbers are based. The user is given the option of listing the generated hemispheric distribution of computed combined P_k's, where the angle alpha denotes azimuth and beta denotes elevation from the burst point to (0,0,0). The range specified is also the range from the burst point to the origin (0,0,0). These hemispheric data (only the positive elevation angles are considered since negative angles would imply a burst below ground) are averaged over all angular bins for which burst points were analyzed to provide a table of average P_k versus range.

The final results are repeated for each case and followed by a summary of the results for each type P_k considered together with the corresponding error data for that case.

After all results have been given for all cases specified, the user is given the opportunity to run additional cases, based upon the same data set. In all cases, the contents of the file TAPE1 are always the last data set considered. Consequently, if the user wishes to make additional runs with ARPSIM at a later time using the same basic data set, then after the current runs with ARPSIM are finished, the file TAPE1 can be saved as a starting point for future runs.

TAPE1 can be retained as a permanent file. However, for access at a later date, this TAPE1 must be attached with a different local file name. Then this local file name is copied to a new file named TAPE1. These steps are necessary because the ARPSIM code changes the contents of the file TAPE1.

APPENDIX C

FORTRAN LISTING

Note: The following FORTRAN listing is subject to changes as dictated by improvements or modifications to the ARPSIM model.

```
      PROGRAM ARP (INPUT=220,OUTPUT=220,TAPE5=INPUT,TAPE6=OUTPUT,
     CTAPE1=220,TAPE2=220,TAPE3=220,TAPE4=220,TAPE8=220)
      DIMENSION ANAM(50),DATA(50),PK1(40,20,8)
      DIMENSION PKS(12,6,10),PKM(50)
      DIMENSION IKS(12,6,10),IKM(50),RGRD(8,41),DGRD(8,21)
      DIMENSION HGT(9),XOMG(3),INEW(50),VTHT(5),GLTR(3,10)
      DIMENSION ALPHA(13),BETA(7),PVT(5)
      DIMENSION PKG(10),CEP(10),RANGE(11),RSUM(10)
      DIMENSION PK(10),PKR(10),PKD(10),PKBL(10)
      DIMENSION RDH(5,2),DDH(5,2),HDH(5,2)
      DIMENSION RRG(10),RRSG(10),BRG(10),BRSG(10),BDG(10),BDSG(10)
      DIMENSION BHG(10),BHSG(10)
      COMMON /SRCH/IPEN,IPN,RBS,DBS,HBS,BR,BD,BH,OMEGA,RPN
     1,DPN,HPN,RDH1,RDH2,DDH1,DDH2,HDH1,HDH2
      COMMON /RDWRT/HGT,XOMG,VTHT,GLTR,PVT,GMR,GMD,
     1 GMH,SDD(10),SDH(10),IDAT(10),RDR(5),BBLST(5),HBLST(5),RU,DU
      DATA ALPHA/0.,30.,60.,90.,120.,150.,180.,210.,240.,270.,300.,
     C330.,360./
      DATA BETA/0.,15.,30.,45.,60.,75.,90./
      DATA DATA/50*0./,YES/1HY/,INEW/50*0/,GLTR/30*0./
      DATA ANAM/4HFZAM,4HFZTM,4HPKDH,4HPKBL,4HFZAS,4HFZTS,4HOMEG,
     C4HNGER,4HNCEP,4HFUNC,4HDHIT,4HPHIM,4HPHIS,4HSAMP,4HRADR,
     C4HDHAZ,4HPKNH,4HPKNA,4HOMGS,4HPDVT,4HPKPF,4HGLTR,4HSRNG,4HPRNT,
     C4HDBUG,4HTGTC,4HDUDR,4HBLST,22*4H    /
      DATA IDAT/8,11,15,17,18,20,22,28,2*0/
      DATA END/4HEND /
      CALL CONNEC (5LTAPE5)
      CALL CONNEC (5LTAPE6)
      REWIND 8
      WRITE (6,2002)
C
      WRITE (6,*) "     ANTI-RADIATION SIMULATION PROGRAM - 1/30/81"
C
      WRITE (6,2002)
      WRITE (6,*) "    NOTE:  ALL COORDINATES ARE DEFINED RELATIVE TO"
      WRITE (6,*) "           ORIGIN AT GROUND ZERO OF TARGET."
C
      WRITE (6,*) "           COORDINATE SYSTEM IS RECTANGULAR,"
      WRITE (6,*) "           TARGET HEADING IS NEGATIVE RANGE,"
      WRITE (6,*) "           DRIVER SIDE (L) IS POSITIVE DEFLECTION,"
      WRITE (6,*) "           HEIGHT IS MEASURED FROM GROUND."
C
C          INEW = 1, NEW DATA THIS CASE.
C
      WRITE (6,2002)
      CALL DATE (DTE)
      CALL TIME (TME)
      WRITE (6,2005) DTE,TME
 2005 FORMAT (//,5X,*DATE - *,A10,/,5X,*TIME - *,A10)
 2002 FORMAT (//,50("*"),/)
 2001 FORMAT (///)
      WRITE (6,2002)
      WRITE (6,*) "DO YOU WANT A LISTING OF CODE NAMES? "
      READ (5,1001) ANS
      IF(ANS.NE.YES) GO TO 54
      WRITE (6,2002)
      WRITE (6,*) "OMEG - ELEVATION OF ATTACK ANGLE"
```

```
      WRITE (6,*) "  NOTE:  FOLLOWING GUIDANCE ERROR PARAMETERS"         000670
      WRITE (6,*) "         (SIGD,SIGH) ARE MEASURED"                    000680
      WRITE (6,*) "         IN PLANE NORMAL TO TRAJECTORY AND"           000690
      WRITE (6,*) "         PASSING THROUGH HOMING POINT"                000700
      WRITE (6,*) "NGER - NUMBER OF GUIDANCE ERRORS TO CONSIDER"         000710
      WRITE (6,*) "         ENTER HOMING POINT (R,D,H), GUIDANCE"        000720
      WRITE (6,*) "         ERRORS ARE DISTRIBUTED ABOUT HOMING PT."     000730
      WRITE (6,*) "NCEP - 1., IF CEP IS INPUT FOR GUIDANCE ERROR SIGMAS"000740
      WRITE (6,*) "FZAM,FZAS,FZTM,FZTS - FUZING ERROR OPTIONS"           000750
      WRITE (6,*) "  FZAM - MEAN ANGLE AT WHICH FUZING OCCURS ON"        000760
      WRITE (6,*) "          INTERCEPT"                                  000770
      WRITE (6,*) "  FZAS - STD DEV ASSOCIATED WITH FZAM"                000780
      WRITE (6,*) "NOTE:  FUZE ANGLE IS CONSTRAINED TO (0,PI)"           000790
      WRITE (6,*) "NOTE:  FOR UNIFORM FUZING ANGLE BETWEEN FZAM"         000800
      WRITE (6,*) "       AND FZAS, ENTER A NEGATIVE VALUE FOR FZAM"     000810
      WRITE (6,*) "       FUZE ANGLE WILL BE CHOSEN UNIFORMLY RANDOM"    000820
      WRITE (6,*) "       BETWEEN POSITIVE FZAM AND FZAS"                000830
      WRITE (6,*) "       FOR TIME-TO-GO FUZE, ENTER NEGATIVE FZAS."     000840
      WRITE (6,*) "NOTE:  FUZING PLANE PASSES THROUGH FUZING GLITTER"    000850
      WRITE (6,*) "       POINT NORMAL TO SAMPLE TRAJECTORY"             000860
      WRITE (6,*) "  FZTM - MEAN DISTANCE FROM GUIDANCE PLANE AT WHICH" 000870
      WRITE (6,*) "          FUZING WILL OCCUR ALONG TRAJECTORY"         000880
      WRITE (6,*) "NOTE:  ENTER A NEGATIVE FZTM FOR HEIGHT FUZING "      000890
      WRITE (6,*) "       WITH MEAN HEIGHT ABS(FZTM) "                   000900
      WRITE (6,*) "  FZTS - STD DEV ASSOCIATED WITH FZTM"                000910
      WRITE (6,*) "SAMP - SAMPLE SIZE"                                   000920
      WRITE (6,*) "PKNH - NUMBER OF HEIGHTS AT WHICH FRAGMENTATION"      000930
      WRITE (6,*) "       PK DATA WILL BE DEFINED"                       000940
      WRITE (6,*) "       NOTE: PKNH < 9"                                000950
      WRITE (6,*) "PKPF - PROBABILITY OF PRIMARY FUZE FUNCTIONING"       000960
      WRITE (6,*) "FDVT - 0. FOR PD BACKUP, NVT FOR VT BACKUP FUZE"      000970
      WRITE (6,*) "       WHERE NVT = NUMBER OF VT BURST HEIGHTS"        000980
      WRITE (6,*) "GLTR - 0. IF PRIMARY FUZE FUNCTIONS RELATIVE TO"      000990
      WRITE (6,*) "       CENTER OF TARGET, NGLT IF PRIMARY FUZE"        001000
      WRITE (6,*) "       FUNCTIONS  RELATIVE TO ANY ONE OF NGLT"        001010
      WRITE (6,*) "       EQUALLY LIKELY GLITTER POINTS ."               001020
      WRITE (6,*) "       SET NGLT NEGATIVE TO PICK FIRST"               001030
      WRITE (6,*) "       POINT ENCOUNTERED."                            001040
      WRITE (6,*) "SRNG - MAXIMUM RANGE FOR COMPUTING PK VS RANGE"       001050
      WRITE (6,*) "FRNT - 1. TO PRINT SUMMARY ONLY, 0. OTHERWISE"        001060
      WRITE (6,*) "DBUG - 6. TO PRINTOUT PROGRAM DEBUGGING DATA"         001070
      WRITE (6,*) "       DBUG = 1, GUIDANCE & FUZING DATA"              001080
      WRITE (6,*) "       DBUG = 2, DIRECT HIT PENETRATION DATA"         001090
      WRITE (6,*) "       DBUG = 4, PK BOX DATA"                         001100
      WRITE (6,*) "       DBUG = 5, PK GRIDS"                            001110
      WRITE (6,*) "       DBUG = 6, PK VS R DATA"                        001120
      WRITE (6,*) "TGTC - HEIGHT OF TARGET CENTER ABOVE GROUND"          001130
      WRITE (6,*) "DUDR - DUD RATE OF PROJECTILE, EXPRESSED AS A FRACTIO001140
     CN"                                                                 001150
      WRITE (6,*) "DHIT - DIRECT HIT OPTION, NUMBER OF TARGET BOXES"     001160
      WRITE (6,*) "       IF DHIT IS OMITTED AND BLST IS INCLUDED,"      001170
      WRITE (6,*) "       BLST IS RADIUS FROM (0,0,TGTC) WITHIN"         001180
      WRITE (6,*) "       WHICH PKBLST = 1."                             001190
      WRITE (6,*) "PKDH - DIRECT HIT PK (0. = 1.)"                       001200
      WRITE (6,*) "PKBL - BLAST PK (0. = 1.)"                            001210
      WRITE (6,*) "RADR - 1., DEFINE FUNC FOR BLAST KILL OF RADAR ONLY" 001220
      WRITE (6,*) "       AND READ IN RADAR ANTENNA COORDINATES."        001230
```

```
      WRITE (6,*) "       TO DEFINE FUNC, SPECIFY R1 AND R2,"
      WRITE (6,*) "       WHERE BLAST PK IS 1 OUT TO R1 AND"
      WRITE (6,*) "       DECLINES LINEARLY TO 0 AT R2."
      WRITE (6,*) "DHAZ - AZIMUTH ANGLE OF ATTACK OFF FRONT OF TARGET"
      WRITE (6,*) "       TOWARD DRIVER SIDF.  SET TO -1. FOR RANDOM"
      WRITE (6,*) "BLST - BLAST RADIUS WITHIN WHICH VEHICLE PK=PKBL"
      WRITE (6,*) "NOTE:  TO ENTER BLAST RADII VS. BURST HEIGHT,"
      WRITE (6,*) "       ENTER NEGATIVE NUMBER OF BLAST,HGT PAIRS"
      WRITE (6,*) "       IN PLACE OF VALUE OF BLST.  PAIRS OF"
      WRITE (6,*) "       BLAST,HGT ARE ENTERED IN ASCENDING ORDER"
      WRITE (6,*) "       OF HEIGHT."
      WRITE (6,*) "COORDINATE SYSTEM IS RECTANGULAR, "
      WRITE (6,*) "TARGET HEADING IS NEGATIVE RANGE, "
      WRITE (6,*) "DRIVER SIDE (LEFT) IS POSITIVE DEFLECTION,"
      WRITE (6,*) "HEIGHT IS MEASURED FROM GROUND"
   54 NPRT = 0
      ISET = 0
      ITIME = 0
      CALL RDMOUT(INIT)
   15 CALL RDMIN(INIT)
      ISET = 1
      IF(IRD.EQ.5) GO TO 88
      IF(NPRT.GT.0) GO TO 80
      WRITE (6,*) "ENTER DATA BY ENTERING CODE NAME"
      WRITE (6,*) "FOLLOWED BY A COMMA AND THE VALUE IN FLOATING"
      WRITE (6,*) "POINT FORMAT.  TO END DATA ENTRY, ENTER "
      WRITE (6,*) "THE WORD END IN COLUMNS 1-3"
C
C          FILE TAPE1 CONTAINS BASIC INPUT DATA
C          FILES TAPE2 - TAPE4 CONTAIN FRAGMENTATION PK GRIDS
C          FOR DIFFERENT ANGLES OF ATTACK
C
   88 WRITE (6,*) "DO YOU WISH TO INITIALIZE DATA FROM"
      WRITE (6,*) "DATA FILE TAPE1?"
      READ (5,1001) ANS
      IRD = 5
      IF(ANS.EQ.YES) IRD = 1
   80 REWIND 1
      REWIND 2
      REWIND 3
      REWIND 4
      PI = ATAN2(0.,-1.)
      DO 51 I=1,10
   51 PKG(I) = 0.
C
C          INITIALIZE OR UPDATE DATA
C
      REWIND 1
    7 IF(IRD.EQ.5) WRITE (6,*) "ENTER DATA OR END - "
      READ (IRD,1000) AAAA,VALUE
 1000 FORMAT (A4,1X,F10.3)
      IF(AAAA.EQ.END) GO TO 14
      DO 53 J=1,50
      IF(AAAA.NE.ANAM(J)) GO TO 53
      INEW(J) = 1
      DATA(J) = VALUE
      GO TO 7
```

```
   53 CONTINUE
      WRITE (6,2000) AAAA
      GO TO 7
   14 CALL READ (DATA,INEW,ANAM,IPD,1,RDH,DDH,HDH)
C
C          SET UP TAPE1
C
    9 REWIND 1
      DO 81 I=1,50
      IF(DATA(I).EQ.0.) GO TO 81
      WRITE (1,1000) ANAM(I),DATA(I)
   81 CONTINUE
      WRITE (1.1000) END
      CALL WRITE (DATA,1,CEP,RDH,DDH,HDH)
      REWIND 1
      IF(ITIME.EQ.0) GO TO 12
      WRITE (6,*) "DO YOU WANT CURRENT INPUT LISTED?"
      READ (5,1001) ANS
      IF(ANS.NE.YES) GO TO 23
      IF(ITIME.GT.0) WRITE (6,*) "CURRENT DATA - "
   12 IF(ITIME.EQ.0) WRITE (6,*) "INITIAL INPUTS - "
C
C          LIST DATA FILE (TAPE1)
C
      DO 8 I=1,50
      READ (1,1000) A,B
      IF(A.EQ.END) GO TO 6
    8 WRITE (6,1002) A,B
 1002 FORMAT (1X,A4,1X,F10.3)
    6 WRITE (6,1002) END
      CALL WRITE (DATA,6,CEP,RDH,DDH,HDH)
   23 REWIND 1
      ITIME = ITIME + 1
      IF(ISET.EQ.1) GO TO 89
      WRITE (6,*) "DO YOU WANT TO CHANGE ANY DATA? - "
      READ (5,1001) ANS
      IF(ANS.NE.YES) GO TO 82
   89 ISET = 0
C
C          READ IN CHANGES
C
      DO 13 I=1,50
   13 INEW(I) = 0
      DO 2 I=1,1000
      WRITE (6,*) "ENTER DATA OR END - "
      READ (5,1000) AAAA,VALUE
      IF(AAAA.EQ.END) GO TO 3
 1001 FORMAT (A1)
      DO 4 J=1,50
      IF(AAAA.NE.ANAM(J)) GO TO 4
      DATA(J) = VALUE
      INEW(J) = 1
      GO TO 2
    4 CONTINUE
      WRITE (6,2000) AAAA
 2000 FORMAT (1X,"***** DO NOT RECOGNIZE ",A4," *****")
    2 CONTINUE
```

```
    3 CALL READ (DATA,INEW,ANAM,5,0,PDH,DDH,HDH)                 002380
      GO TO 9                                                    002390
   82 DO 83 I=1,50                                               002400
   83 INEW(I) = 0                                                002410
C                                                                002420
C          SET UP DATA                                           002430
C                                                                002440
C          LOAD INPUT DATA INTO VARIABLE SET                     002450
C          AND CONVERT DEGREES TO RADIANS                        002460
C                                                                002470
      FZAM = DATA(1)/57.29578                                    002480
      FZTM = ABS(DATA(2))                                        002490
      PKDHX = DATA(3)                                            002500
      PKBLX = DATA(4)                                            002510
      FZAS = DATA(5)/57.29578                                    002520
      ITTG = 0                                                   002530
      IF(FZAS.LT.0.) ITTG = 1                                    002540
      FZAS = ABS(FZAS)                                           002550
      FZTS = DATA(6)                                             002560
      OMEG = DATA(7)/57.29578                                    002570
      NGER = DATA(8)                                             002580
      NCEP = DATA(9)                                             002590
      IFUN = 0                                                   002600
      NDHT = DATA(11)                                            002610
      NSMP = DATA(14)                                            002620
      NRDR = DATA(15)                                            002630
      DHAZ = DATA(16)/57.29578                                   002640
      NH = DATA(17)                                              002650
      NA = 0                                                     002660
      OMGS = 0.                                                  002670
      PKPF = DATA(21)                                            002680
      NVT = DATA(20)                                             002690
      NGLT = DATA(22)                                            002700
      JGLT = 1                                                   002710
      JGLT = ISIGN(JGLT,NGLT)                                    002720
      NGLT = IABS(NGLT)                                          002730
      SRNG = DATA(23)                                            002740
      NPRT = DATA(24)                                            002750
      NDBG = DATA(25)                                            002760
      TGTC = DATA(26)                                            002770
      DUDR = DATA(27)                                            002780
      BLST = DATA(28)                                            002790
      IF(BLST.LE.0.) GO TO 94                                    002800
      BBLST(1) = BLST                                            002810
      HBLST(1) = 100000.                                         002820
      BLST = 1.                                                  002830
   94 NBLST = ABS(BLST)                                          002840
      IHFZ = 0                                                   002850
      IF(DATA(2).LT.0.) IHFZ = 1                                 002860
      IF(PKDHX.EQ.0.) PKDHX = 1.                                 002870
      IF(PKBLX.EQ.0.) PKBLX = 1.                                 002880
      NLOOP = NGER                                               002890
      IF(NDBG.GE.1) WRITE (6,*) "DBUG OPTION ",NDBG              002900
      IF(DATA(2).NE.0.) IFUZ = 2                                 002910
      IF(DATA(1).NE.0.) IFUZ = 1                                 002920
      XRNG = 0.                                                  002930
      IF(NDHT.EQ.0) GO TO 115                                    002940
```

```
      DO 116 I=1,NDHT
      RRNG = 10000.
      IF(SIGN(1.,DDH(I,1)).EQ.SIGN(1.,DDH(I,2))) GO TO 118
      IF(SIGN(1.,RDH(I,1)).EQ.SIGN(1.,RDH(I,2))) GO TO 118
      DO 119 J=1,2
      RRNG = AMIN1(RRNG,DDH(I,J))
  119 RRNG = AMIN1(RRNG,RDH(I,J))
      XRNG = SQRT(RRNG**2. + HDH(I,1)**2.)
      GO TO 116
  118 XRNG = HDH(I,1)
  116 CONTINUE
      XRNG = AMIN1(XRNG,HDH(NDHT,2))
  115 IF(SRNG.EQ.0.) SRNG = 100.
      DL = ALOG(SRNG-XRNG)/10.
      DO 111 I=1,10
      XI = I
  111 RANGE(I) = XRNG + EXP(DL*XI)
      RANGE(11) = 1000.
      IF(NVT.LE.1) GO TO 67
      DO 68 I=2,NVT
   68 PVT(I) = PVT(I) + PVT(I-1)
   67 IF(NGLT.GT.0) GO TO 59
      DO 60 I=1,3
   60 GLTR(I,1) = 0.
   59 IF(NA.EQ.0) GO TO 48
      DO 28 I=1,NA
   28 XOMG(I) = XOMG(I)/57.29578
   48 CONTINUE
C
C         READ IN PK GRIDS FOR EACH ATTACK ANGLE,BURST HEIGHT
C         COMBINATION
C
      IF(NH.EQ.0) GO TO 78
      CALL GRIDS (PK1,NH,2,RGRD,DGRD,NR,ND,NDEG)
C
C         LOOP OVER SIMULATIONS FOR EACH GUIDANCE ERROR SET
C
   78 DO 69 ILUP=1,NLOOP
C
C         INITIALIZE COUNTERS
C
      DO 70 I=1,50
      PKM(I) = 0.
   70 IKM(I) = 0
      DO 52 I=1,12
      DO 52 J=1,6
      DO 52 K=1,10
      IKS(I,J,K) = 0
   52 PKS(I,J,K) = 0.
      PKRADR = 0.
      PKDHIT = 0.
      PKBASE = 0.
      PKBLT = 0.
      PKTOT = 0.
      PKTOT2 = 0.
      RRBAR = 0.
      RRBAR2 = 0.
```

```
      BDBAR = 0.
      BDBAR2 = 0.
      BRBAR = 0.
      BRBAR2 = 0.
      BHBAR = 0.
      BHBAR2 = 0.
      IF(PKPF.EQ.0.) PKPF = 1.
      IF(PKPF.LT.0.) PKPF = 0.
      SIGD = SDD(ILUP)
      SIGH = SDH(ILUP)
      NCT = 0
C
C          BEGIN SIMULATIONS
C
      DO 1 ISIM=1,NSMP
      IF(DATA(16).LT.0.) DHAZ = RDM(1)*2.*PI
      PKSAMP = 0.0
      PKDH = 0.
      PKBLST = 0.
      PKRDR = 0.
C
C          CHECK FOR DUD
C
      IF(RDM(1).LE.DUDR) GO TO 18
C
C          SAMPLE FROM ATTACK ANGLE DISTRIBUTION
C
      CALL BOXNO (Z1,Z2)
      OMEGA = Z1*OMGS + OMEG
      SINO = SIN(OMEGA)
      COSO = COS(OMEGA)
      TANO = 1.
      IF(COSO.NE.0.) TANO = SINO/COSO
C
C          ROTATE COORDINATES OF HOMING POINT ACCORDING
C          TO AZIMUTH COMPONENT OF ATTACK ANGLE.
C
C          ALL COMPUTATIONS TO DETERMINE FUZING POINT ARE IN
C          ROTATED COORDINATE SYSTEM.
C
C
      GMRR = GMR
      GMDR = GMD
      CALL ROTATE (GMRR,GMDR,DHAZ,1.)
C
C          SAMPLE FROM GUIDANCE ERROR DISTRIBUTION
C          RELATIVE TO HOMING POINT
C
      CALL BOXNO (D,H)
      DMIN = SQRT((SIGH*H)**2. + (SIGD*D)**2.)
      GR = GMRR + SIGH*H*SINO
      GD = GMDR + SIGD*D
      GH = GMH + SIGH*H*COSO
C
C          (GR,GD,GH) IS INTERCEPT OF
C          TRAJECTORY WITH GUIDANCE PLANE
C          (RF,DF,HF) WILL BE FUZING POINT ON TRAJECTORY.
```

```
      C
            RF = GR
            DF = GD
            HF = GH
      C
      C          CHECK FOR PRIMARY FUZE FUNCTION
      C
            IBKUP = 0
            IF(RDM(1).GT.PKPF) GO TO 16
      C
      C          CHECK FOR HEIGHT FUZING
      C
            IF(IHFZ.EQ.1) GO TO 74
            D2 = 0.
      C
      C          CHECK FOR APPROPRIATE FUZING
      C
            CALL BOXNG (Z1,Z2)
            IGO = IFUZ + 1
            IF(NDBG.GE.1) WRITE (6,5003) IFUZ,IGO,GR,GD,GH
            GO TO (55,75,22,85),IGO
      C
      C          CHOSE GLITTER POINT FOR FUZING, ANGULAR FUZE ONLY
      C
         75 IF(JGLT.LT.0.AND.NGLT.GT.1) GO TO 76
            XGLT = NGLT
            IGLT = (RDM(1)-0.0001)*XGLT + 1.0
            IF(IGLT.EQ.0) IGLT = 1
         86 RGLT = GLTR(1,IGLT)
            DGLT = GLTR(2,IGLT)
            HGLT = GLTR(3,IGLT)
            IDO = 1
            GO TO 77
         76 IDO = NGLT
            GRMAX = -100000.
         77 DO 84 IGL=1,IDO
            IF(IDO.EQ.1) GO TO 21
            RGLT = GLTR(1,IGL)
            DGLT = GLTR(2,IGL)
            HGLT = GLTR(3,IGL)
         21 IF(NDBG.EQ.1) WRITE (6,*) "RGLT,DGLT,HGLT = ",RGLT,DGLT,HGLT
      C
      C          ROTATE GLITTER POINT INTO ARP COORDINATE SYSTEM
      C
            CALL ROTATE (RGLT,DGLT,DHAZ,1.)
            IF(NDBG.EQ.1) WRITE (6,*) "ROTATED GLITTER POINT = "
            IF(NDBG.EQ.1) WRITE (6,*) "DHAZ,RGLT,DGLT,HGLT = ",DHAZ,RGLT,DGLT,
           C HGLT
       5003 FORMAT (1X,*IFUZ,IGO = *,2(I2,*,*,1X),*GR,GD,GH = *,3(F6.1,*,*,1X))
          C
      C
      C          USE LAW OF SINES AND LAW OF COSINES TO FIND
      C          FUZING POINT ON TRAJECTORY.  FIRST PICK A POINT
      C          ALONG TRAJECTORY TO COMPUTE BETAX (ANGLE BETWEEN
      C          TRAJECTORY AND A LINE (AB) FROM GLITTER POINT
      C          (RGLT,DGLT,HGLT) TO GUIDANCE PLANE INTERCEPT
      C          (GR,GD,GH) - NOTE THAT EVERYTHING IS IN ROTATED
```

```
C             COORDINATE SYSTEM (THROUGH AZIMUTH ATTACK ANGLE
C             COMPONENT).  THEN, KNOWING BETAX AND FUZING ANGLE
C             (ANG) COMPUTE ANGLE (GAMMA) WITH ITS VERTEX AT
C             GLITTER POINT AND OPPOSITE TRAJECTORY SEGMENT
C             BOUNDED BY GUIDANCE PLANE INTERCEPT AND FUZING
C             POINT.  FINALLY, KNOWING GAMMA, AB, AND ANG, COMPUTE
C             O2, THE DISTANCE FROM GUIDANCE PLANE INTERCEPT
C             TO FUZING POINT (USING THE LAW OF SINES).
C
          TANOX = TANO
          IF(SINO.EQ.0.) TANOX = 1.
          CB = 10.
          IF(SINO.NE.0.) CB = CB/SINO
C
C             GRL,GDL,GHL ARE COORDINATES OF A POINT ON
C             THE TRAJECTORY USED TO COMPUTE BETAX.
C
          GRL = GR - 10./TANOX
          GDL = GD
          GHL = GH
          IF(SINO.NE.0.) GHL = GH + 10.
          AB2 = (RGLT-GR)**2. + (DGLT-GD)**2. + (HGLT-GH)**2.
          BB2 = (RGLT-GRL)**2. + (DGLT-GDL)**2. + (HGLT-GHL)**2.
          AB = SQRT(AB2)
C
C             USE LAW OF COSINES TO COMPUTE BETAX, ANGLE WITH VERTEX AT
C             GLITTER POINT AND OPPOSITE TRAJECTORY SEGMENT
C             BOUNDED BY GUIDANCE PLANE INTERCEPT AND FUZING POINT.
C
          BETAX = ACOS((AB2-BB2+CB*CB)/(2.*AB*CB))
          IF(NDBG.EQ.1) WRITE (6,*) "BETAX,GRL,GDL,GHL,AB,CB = ",
        C BETAX,GRL,GDL,GHL,AB,CB
          FZASX = FZAS
          IF(ITTG.EQ.1) FZASX = 0.
C
C             ANGULAR FUZING FUNCTION
C
          ANG = Z2*FZASX + FZAM
          IF(FZAM.LT.0.) ANG = FZAM + RDM(1)*(FZASX-FZAM)
          IF(ANG.LT..01745) GO TO 18
          IF(ANG.GT.PI) GO TO 16
C
C             O2 IS DISTANCE ALONG TRAJECTORY FROM GUIDANCE
C             PLANE INTERCEPT TO FUZING POINT.
C
          GAMMA = PI - BETAX - ANG
C
C             IF GAMMA.LT.ZERO, USE SUPPLEMENT OF ANG FOR FUZING.
C
          IF(GAMMA.LT.0.) ANG = PI - ANG
          O2 = AB*(SIN(GAMMA)/SIN(ANG))
          IF(NDBG.EQ.1) WRITE (6,*) "O2,GAMMA,ANG = ",O2,GAMMA,ANG
          IF(IDO.EQ.1) GO TO 22
          IF(O2.LT.GRMAX) GO TO 84
          GRMAX = O2
          IGLT = IGL
       84 CONTINUE
```

```
      GO TO 86
C
C           LINEAR FUZING FUNCTION (ALONG TRAJECTORY)
C           FUZING DIRECTION IS POSITIVE IN THE NEGATIVE
C           RANGE DIRECTION, I.E., A POSITIVE CHANGE IN
C           THE FUZING DISTANCE, D2, IS IN THE NEGATIVE
C           RANGE DIRECTION.
C
   22 IF(ITTG.EQ.1) FZTS = DMIN*TAN(FZAS)
      D2 = D2 + Z2*FZTS + FZTM
      RF = GR - D2*COSO
      HF = GH + D2*SINO
      DF = GD
      GO TO 85
C
C           BACKUP FUZING
C
   16 HF = 0.
      IBKUP = 1
      IF(OMEGA.EQ.0.) GO TO  5
      IF(NVT.EQ.0) GO TO 17
   87 XK = RDM(1)
      DO 65 K=1,NVT
      KK = K
      IF(XK.LE.PVT(K)) GO TO 66
   65 CONTINUE
   66 HFX = VTHT(KK)
      IF(HFX.LE.HF) GO TO 24
      HF = HFX
   17 RF = GR - (HF-GH)/TANO
      DF = GD
      GO TO 61
    5 WRITE (6,*) "NO BACKUP FUZING FOR OMEGA = 0."
      WRITE (6,*) "TRAJECTORY CLOSEST POINT OF APPROACH TO TARGET"
      WRITE (6,*) "CENTER IS USED"
      RF = 0.
      DF = GD
      HF = GH
      GO TO 61
C
C           HEIGHT FUZING
C
   74 IF(SINO.EQ.0.) STOP 74
      CALL BOXNO (Z1,Z2)
      HF = FZTM + Z1*FZTS
      RF = RF + (GH-HF)/TANO
   85 IF(OMEGA.EQ.0.) GO TO 24
      IF(NVT.NE.0) GO TO 87
C
C           CHECK FOR FUZING POINT BELOW GROUND
C
   24 IF(HF.GE.0.) GO TO 61
      IF(OMEGA.EQ.0.) GO TO 61
      RF = RF + HF/TANO
      HF = 0.
C
C           PUT BURST POINT IN TARGET COORDINATE SYSTEM FOR
```

```
C          BLAST AND DIRECT HIT COMPUTATIONS.
C
    61 CALL ROTATE (RF,DF,DHAZ,-1.)
       BR = RF
       BD = DF
       BH = HF
       IF(NDBG.GE.1) WRITE (6,*) "BR,BD,BH AT STMT 61 = ",BR,BD,BH
C
C          SET UP BLST VALUE FOR BLST VS. HGT
C
       IF(NBLST.LE.0) GO TO 105
       DO 10 I=1,NBLST
       IF(HF.GT.HBLST(I)) GO TO 10
       BLST = BBLST(I)
       GO TO 105
    10 CONTINUE
       BLST = 0.
       WRITE (6,*) "HF EXCEEDS ALL HBLST, HF = ",HF
       GO TO 18
   105 IF(NDHT.EQ.0) GO TO 106
C
C          DETERMINE DIRECT HIT PK
C
C
C          USE 2 POINTS TO DEFINE TRAJECTORY, BURST POINT
C          (BR,BD,BH) AND POINT AT BR+10 (RBS,DBS,HBS).
C          IF AZIMUTH ATTACK ANGLE IS 90 DEGREES, SET
C          RBS,DBS,HBS POINT AT BD+10.
C          (RPN,DPN,HPN) WILL BE BURST POINT, WITH OR
C          WITHOUT DIRECT HIT.
C
C          IPN IS PENETRATION INDEX (0 = NO PENETRATION,
C          N = BOX N PENETRATED)
C
       RPN = BR
       DPN = BD
       HPN = BH
       IF(ABS(DATA(16)).EQ.90.) GO TO 95
       RBS = BR + 10.
       DBS = BD - 10.*TAN(DHAZ)
       HBS = BH - 10.*TANO/COS(DHAZ)
       GO TO 96
    95 RBS = BR
       DBS = BD + 10.
       HBS = BH + 10.*TANO
    96 IPN = 0
C
C          CHECK EACH BOX FOR PENETRATION
C
       IF(NDBG.EQ.1) WRITE (6,*) "OMEGA,RBS,DBS,HBS = ",OMEGA,RBS,DBS,HBS
       IF(NDBG.EQ.1) WRITE (6,*) "RF,DF,HF = ",RF,DF,HF
       IF(NDBG.EQ.1) WRITE (6,*) "GR,GD,GH = ",GR,GD,GH
       DO 92 I=1,NDHT
       IF(BR.LT.RDH(I,1)) GO TO 92
       IF(DATA(16).NE.0.) GO TO 109
       IF(BD.LT.DDH(I,1).OR.BD.GT.DDH(I,2)) GO TO 92
   109 IF(BH.GT.HDH(I,2).AND.OMEGA.GE.0.) GO TO 92
```

```
      IF(BH.LT.HDH(I,1).AND.OMEGA.EQ.0.) GO TO 92                    006370
      RDH1 = RDH(I,1)                                                006380
      RDH2 = RDH(I,2)                                                006390
      DDH1 = DDH(I,1)                                                006400
      DDH2 = DDH(I,2)                                                006410
      HDH1 = HDH(I,1)                                                006420
      HDH2 = HDH(I,2)                                                006430
C                                                                    006440
C          IPEN = NUMBER OF SIDES PENETRATED (MUST BE 0 OR 2)        006450
C                                                                    006460
      IPEN = 0                                                       006470
      IF(ABS(DATA(16)).EQ.90.) GO TO 102                             006480
C                                                                    006490
C          CHECK RANGE SIDES                                         006500
C                                                                    006510
      DO 97 K=1,2                                                    006520
      RDHX = RDH1                                                    006530
      IF(K.EQ.2) RDHX = RDH2                                         006540
      CALL SEARCH (I,1,RDHX,DA,HA)                                   006550
      IF(NDBG.EQ.2) WRITE (6,*) "IPEN,RDHX,DA,HA = ",IPEN,           006560
     1 RDHX,DA,HA                                                    006570
   97 CONTINUE                                                       006580
      IF(IPEN.EQ.2) GO TO 92                                         006590
  102 IF(DATA(16).EQ.0..OR.DATA(16).EQ.180.) GO TO 108               006600
C                                                                    006610
C          CHECK DEFLECTION SIDES                                    006620
C                                                                    006630
      DO 107 K=1,2                                                   006640
      DDHX = DDH1                                                    006650
      IF(K.EQ.2) DDHX = DDH2                                         006660
      CALL SEARCH (I,2,RA,DDHX,HA)                                   006670
      IF(NDBG.EQ.2) WRITE (6,*) "IPEN,RA,DDHX,HA = ",IPEN,           006680
     1 RA,DDHX,HA                                                    006690
      IF(IPEN.EQ.2) GO TO 92                                         006700
  107 CONTINUE                                                       006710
  108 IF(OMEGA.EQ.0.) GO TO 101                                      006720
C                                                                    006730
C          CHECK HEIGHT SIDES                                        006740
C                                                                    006750
      DO 117 K=1,2                                                   006760
      HDHX = HDH1                                                    006770
      IF(K.EQ.2) HDHX = HDH2                                         006780
      CALL SEARCH (I,3,RA,DA,HDHX)                                   006790
      IF(NDBG.EQ.2) WRITE (6,*) "IPEN,RA,DA,HDHX = ",IPEN,           006800
     1 RA,DA,HDHX                                                    006810
      IF(IPEN.EQ.2) GO TO 92                                         006820
  117 CONTINUE                                                       006830
  101 IF(IPEN.EQ.1) STOP 117                                         006840
   92 CONTINUE                                                       006850
      IF(IPN.EQ.0) GO TO 106                                         006860
      PKDH = PKDH + PKDHX                                            006870
C                                                                    006880
C          SET UP BURST COORDINATES (BR,BD,BH) FROM DIRECT HIT.      006890
C                                                                    006900
      BR = RPN                                                       006910
      BD = DPN                                                       006920
      BH = HPN                                                       006930
```

```
  106 IF(BH.GE.0.) GO TO 37
      IF(OMEGA.EQ.0.) STOP 106
      BR = BR + BH/TANO
      BH = 0.
C
C          COMPUTE NEAR MISS BLAST KILL
C
   37 IF(NBLST.EQ.0) GO TO 90
      IF(NDHT.EQ.0) GO TO 103
      DO 104 I=1,NDHT
      IBLST = 1
      CALL BLAST (IBLST,BR,BLST,RDH,I)
      CALL BLAST (IBLST,BD,BLST,DDH,I)
      CALL BLAST (IBLST,BH,BLST,HDH,I)
      IF(IBLST.EQ.1) GO TO 11
  104 CONTINUE
      GO TO 90
  103 DIST = SQRT(BR*BR + BD*BD + (BH-TGTC)*(BH-TGTC))
      IF(DIST.GT.BLST) GO TO 90
   11 PKBLST = PKBLST + PKBLX
C
C          COMPUTE RADAR BLAST KILL
C
   90 IF(NDBG.EQ.2) WRITE (6,*) "IPN,RPN,DPN,HPN,BR,BD,BH = ",
     C IPN,RPN,DPN,HPN,BR,BD,BH
      IF(NRDR.EQ.0) GO TO 27
      BRDR = BR-RDR(1)
      DRDR = BD-RDR(2)
      HRDR = BH-RDR(5)
      RRDR = SQRT(BRDR*BRDR+DRDR*DRDR+HRDR*HRDR)
      PKRDR = 1.0
      IF(RRDR.GT.RDR(4)) PKRDR = 1. - (RRDR-RDR(4))/(RDR(5)-RDR(4))
      IF(RRDR.GE.RDR(5)) PKRDR = 0.
 5004 FORMAT (1X,*BR,BD,BH = *,3(F6.1,*,*,1X))
   27 IBX = 0
      IROT = 0
      IF(NDBG.GE.1) WRITE (6,5004) BR,BD,BH
      IF(NH.EQ.0) GO TO 50
C
C          COMPUTE PK DUE TO FRAGMENTATION (PKSAMP)
C
C
C          INTERPOLATE IN RANGE, DEFLECTION, HEIGHT & ANGLE TO
C          GET FRAGMENTATION PK FROM PK GRIDS.
C
      II = 1
      IF(BH.GT.HGT(NH+1)) GO TO 50
C
C          ROTATE BURST POINT FOR FRAGMENTATION PK
C          INTERPOLATION INTO ARP COORDINATE SYSTEM.
C          RECALL THAT PK GRIDS ARE IN PROJECTILE COORDINATE
C          SYSTEM.
C
      CALL ROTATE (BR,BD,DHAZ,1.)
      IROT = 1
C
C          LOCATE HEIGHT BOUNDARIES
```

```
C
      DO 20 I=1,NH
      IH2 = I
      IF(BH.LE.HGT(I)) GO TO 25
   20 CONTINUE
      IH2 = 0
   25 IH1 = IH2 - 1
      IF(IH1.EQ.0) IH1 = 1
      IF(IH1.LT.0) IH1 = NH
      IF(NDBG.EQ.4) WRITE (6,*) "IH1,IH2,NR,ND,RU,DU,BR,BD,BH = ",
     C IH1,IH2,NR,ND,RU,DU,BR,BD,BH
   31 CALL INTERP(BR,BD,BH,RGRD,CGRD,HGT,IH1,IH2,PK1,PKA,NR,ND,RU,DU,NH,
     C NDBG)
      PKSAMP = PKA
      GO TO 41
   50 PKSAMP = 0.
C
C           COMPUTE SPHERICAL COORDINATES TO BURST POINT (BR,BD,BH)
C           FROM GROUND ZERO  (0,0,0)
C           SA1 = ANGLE OFF POSITIVE RANGE AXIS MEASURED
C           CLOCKWISE
C           SA2 = ANGLE OFF R-D PLANE MEASURED TOWARD POSITIVE
C           SR = RANGE FROM BURST POINT TO (0,0,0)
C           H-AXIS IN VERTICLE PLANE
C
   41 IF(NDBG.EQ.4) WRITE (6,*) "PK(FRAG) = ",PKSAMP
C
C           GET BURST POINT BACK INTO TARGET COORDINATE
C           SYSTEM IF IROT = 1.
C
      IF(IROT.EQ.1) CALL ROTATE (BR,BD,DHAZ,-1.)
      BRR = BR*BR
      BDD = BD*BD
      BHH = BH*BH
      RRR = BRR + BDD + BHH
      RR = SQRT(RRR)
      WRITE (8,*) BR,BD,BH,RR
      BRBAR = BRBAR + BR
      BRBAR2 = BRBAR2 + BRR
      BDBAR = BDBAR + BD
      BDBAR2 = BDBAR2 + BDD
      BHBAR = BHBAR + BH
      BHBAR2 = BHBAR2 + BHH
      RRBAR = RRBAR + RR
      RRBAR2 = RRBAR2 + RRR
      SA1 = PI/2.
      SA2 = 0.
      IF(BR.EQ.0.) GO TO 55
      SA1 = ATAN2(BD,BR)
      IF(SA1.LT.0.) SA1 = 2.*PI + SA1
   55 IF(BD.EQ.0..AND.BR.EQ.0.) GO TO 56
      SA2 = ATAN(BH/SQRT(BR*BR+BD*BD))
   56 SA1 = SA1*360./(2.*PI)
      SA2 = SA2*360./(2.*PI)
      DO 57 I=1,12
      ISA1 = I
      IF(SA1.LT.ALPHA(I+1)) GO TO 58
```

```
   57 CONTINUE
   58 DO 98 I=1,6
      ISA2 = I
      IF(SA2.LT.BETA(I+1)) GO TO 99
   98 CONTINUE
   99 SR = SQRT(BR*BR + BD*BD + BH*BH)
      ISR = 0
      DO 100 I=1,10
      II = I
      IF(I.EQ.10) II = 11
      ISR = ISR + 1
      IF(SR.LT.RANGE(II)) GO TO 110
  100 CONTINUE
  110 IF(NDBG.EQ.6) WRITE (6,*) "ISA1,ISA2,ISR = ",ISA1,ISA2,ISR
      IF(NDBG.EQ.6) WRITE (6,*) "SA1,SA2,SR = ",SA1,SA2,SR
C
C          STORE PK'S ACCORDING TO SPHERICAL COORDINATES
C
      IKS(ISA1,ISA2,ISR) = IKS(ISA1,ISA2,ISR) + 1
C
C          SUM PK'S OVER ALL SAMPLES
C
      IF(NDBG.GT.0) WRITE (6,*) "PKR,PKR,PKD,PKB = ",PKSAMP,PKRDR,PKDH
     C,PKBLST
      PKBASE = PKBASE + PKSAMP
      PKRADR = PKRADR + PKRDR
      PKDHIT = PKDHIT + PKDH
      PKBLT = PKBLT + PKBLST
      PKSAMP = 1. - (1.-PKSAMP)*(1.-PKRDR)*(1.-PKDH)*(1.-PKBLST)
      PKS(ISA1,ISA2,ISR) = PKS(ISA1,ISA2,ISR) + PKSAMP
      PKTOT = PKTOT + PKSAMP
      PKTOT2 = PKTOT2 + PKSAMP*PKSAMP
      IF(NDBG.GE.1) WRITE (6,3003) PKSAMP
 3003 FORMAT (5X,*SAMPLE PK = *,F6.4)
      IF(NPRT.EQ.1) GO TO 1
      IF(MOD(ISIM,10).NE.0) GO TO 1
      PKPRNT = ISIM
      PKPRNT = PKTOT/PKPRNT
      WRITE (6,*) "NO. SIMULATIONS, PK = ",ISIM,PKPRNT
      GO TO 1
   18 NCT = NCT + 1
    1 CONTINUE
C
C          DISPLAY FINAL RESULTS
C
      IF(NPRT.GT.0) GO TO 79
      WRITE (6,2002)
      WRITE (6,*) "FINAL RESULTS"
 3000 FORMAT (/,1X,*PK = *,F6.4,2X,*PKSD = *,F6.4,2X,*NSAMP = *,I6,/)
   79 XSAMP = NSMP
      PKBAR = PKTOT/XSAMP
      PKBASE = PKBASE/XSAMP
      PKRADR = PKRADR/XSAMP
      PKDHIT = PKDHIT/XSAMP
      PKBLT = PKBLT/XSAMP
      PK(ILUP) = PKBASE
      PKR(ILUP) = PKRADR
```

```
          PKD(ILUP) = PKDHIT                                                008650
          PKG(ILUP) = PKBAR                                                 008660
          PKBL(ILUP) = PKBLT                                                008670
          XSAMP = NSMP - NCT                                                008680
          IF(NSMP.EQ.NCT) XSAMP = 1.                                        008690
          XSMP = XSAMP-1.                                                   008700
          IF(XSMP.EQ.0.) XSMP = 1.                                          008710
          BRG(ILUP) = BRBAR/XSAMP                                           008720
          BDG(ILUP) = BDBAR/XSAMP                                           008730
          BHG(ILUP) = BHBAR/XSAMP                                           008740
          RRG(ILUP) = PRBAR/XSAMP                                           008750
          BRSG(ILUP) = SQRT((BRBAR2 - XSAMP*BRG(ILUP)*BRG(ILUP))/XSMP)      008760
          BDSG(ILUP) = SQRT((BDBAR2 - XSAMP*BDG(ILUP)*BDG(ILUP))/XSMP)      008770
          BHSG(ILUP) = SQRT((BHBAR2 - XSAMP*BHG(ILUP)*BHG(ILUP))/XSMP)      008780
          RRSG(ILUP) = SQRT((RRBAR2 - XSAMP*RRG(ILUP)*RRG(ILUP))/XSMP)      008790
          IF(NCT.NE.0) WRITE (6,2004) NCT,NSMP,CEP(ILUP)                    008800
     2004 FORMAT (2X,*PROJECTILE OR FUZING DUDS = *,I4,* OUT OF *,I4,       008810
         C * SIMULATIONS, *,/.2X,*GUIDANCE CEP = *,F6.2)                    008820
          IF(NPRT.GT.0) GO TO 69                                            008830
          PKSD = (PKTOT2 - XSAMP*PKBAR*PKBAR)/XSMP                          008840
          IF(PKSD.LT.0.) PKSD = 0.                                          008850
          PKSD = SQRT(PKSD)                                                 008860
          WRITE (6,3000) PKBAR,PKSD,NSMP                                    008870
          WRITE (6,2002)                                                    008880
          WRITE (6,*) "DO YOU WANT PK VS R. ALPHA, BETA? "                  008890
          READ (5,1001) ANS                                                 008900
          IF(ANS.NE.YES) GO TO 44                                           008910
C                                                                           008920
C              PK VS R, ALPHA, BETA, WHERE ALPHA IS AZIMUTH ANGLE           008930
C              MEASURED FROM POSITIVE RANGE AXIS TOWARD POSITIVE            008940
C              DEFLECTION AXIS (0 TO 360), BETA IS ELEVATION ANGLE          008950
C              MEASURED FROM NEGATIVE HEIGHT AXIS TO POSITIVE               008960
C              HEIGHT AXIS (0 TO 90).                                       008970
C                                                                           008980
          WRITE (6,2001)                                                    008990
          WRITE (6,*) "  PK       R         ALPHA              BETA"        009000
          WRITE (6,*) "------   -----   --------------   --------------"    009010
          DO 49 I=1,10                                                      009020
          DO 49 J=1,12                                                      009030
          DO 49 K=1,6                                                       009040
          IF(IKS(J,K,I).EQ.0) GO TO 49                                      009050
          XIKS = IKS(J,K,I)                                                 009060
          PKS(J,K,I) = PKS(J,K,I)/XIKS                                      009070
       49 CONTINUE                                                          009080
          DO 45 I=1,10                                                      009090
          XI = 0.                                                           009100
          RSUM(I) = 0.                                                      009110
          RANG = RANGE(I)                                                   009120
          DO 47 J=1,12                                                      009130
          DO 47 K=1,6                                                       009140
          RPK = PKS(J,K,I)                                                  009150
          XIKS = IKS(J,K,I)                                                 009160
          XI = XI + XIKS                                                    009170
          RSUM(I) = RSUM(I) + XIKS*RPK                                      009180
          IF(RPK.GT.0.) WRITE (6,3004) RPK,RANG,ALPHA(J),ALPHA(J+1),BETA(K),009190
         CBETA(K+1)                                                         009200
     3004 FORMAT (1X,F6.4,2X,F5.1,2(2X,F6.1,* - *,F6.1))                    009210
```

```
   47 CONTINUE                                                          009220
      IF(XI.EQ.0.) GO TO 45                                             009230
      RSUM(I) = RSUM(I)/XI                                              009240
   45 CONTINUE                                                          009250
      WRITE (6,2002)                                                    009260
      WRITE (6,*) "AVG PK VS. R"                                        009270
      WRITE (6,*) "------------"                                        009280
      DO 43 I=1,10                                                      009290
      R = RANGE(I)                                                      009300
      IF(RSUM(I).EQ.0.) GO TO 43                                        009310
      WRITE (6,3001) RSUM(I),R                                          009320
   43 CONTINUE                                                          009330
 3001 FORMAT (1X,F6.4,4X,F5.1)                                          009340
      WRITE (6,2002)                                                    009350
C          CHECK FOR ANOTHER CASE                                       009360
C                                                                       009370
   44 WRITE (6,2001)                                                    009380
   69 CONTINUE                                                          009390
C                                                                       009400
C          DISPLAY RESULTS FOR EACH GUIDANCE ERROR                      009410
C                                                                       009420
      IF(NPRT.GT.0) WRITE (6,2002)                                      009430
      FZTM = DATA(2)                                                    009440
      OMEGD = DATA(7)                                                   009450
      OMGSD = DATA(19)                                                  009460
      FZAMD = DATA(1)                                                   009470
      FZASD = DATA(5)                                                   009480
      WRITE (6,2006) OMEGD,OMGSD,FZAMD,FZASD,FZTM,FZTS,DHAZ,NSMP        009490
 2006 FORMAT (/,5X,*RESULTS FOR FOLLOWING CONDITIONS - *,//,            009500
     C12X,*ITEM*,13X,*MEAN*,4X,*STD DEV*,//,                            009510
     C10X,*ELEVATION*,4X,2F10.4,/,10X,*FUZE ANGLE*,3X,2F10.4,/,         009520
     C10X,*LINEAR FUZE*,2X,2F10.4,/,10X,*AZIMUTH      *,F10.4,/,        009530
     C10X,*SAMPLE SIZE - *,I5,/)                                        009540
      WRITE (6,2003) GMR,GMD,GMH                                        009550
      WRITE (6,2012)                                                    009560
 2012 FORMAT (/,5X,*ERROR DATA*,17X,*PK*,3X,                            009570
     C*PKFRAG PKRADR PKDHIT PKBLST*)                                    009580
 2003 FORMAT (5X,*HOMING POINT COORDINATES (R,D,H) = *,                 009590
     C 2(F6.1,*, *),F6.1)                                               009600
      DO 72 I=1,NLOOP                                                   009610
      IF(NCEP.EQ.0) WRITE (6,2007) SDD(I),SDH(I),PKG(I)                 009620
     C,PK(I),PKR(I),PKD(I),PKBL(I)                                      009630
      IF(NCEP.EQ.1) WRITE (6,2008) CEP(I),PKG(I)                        009640
     C,PK(I),PKR(I),PKD(I),PKBL(I)                                      009650
   72 CONTINUE                                                          009660
      WRITE (6,2002)                                                    009670
      WRITE (6,1003)                                                    009680
      DO 26 I=1,NLOOP                                                   009690
   26 WRITE (6,1004) CEP(I),RRG(I),RRSG(I),BRG(I),BRSG(I),BDG(I),BDSG(I)009700
     C ,BHG(I),BHSG(I)                            1            PROGRAM ARP       73/74   OPT=1

      PROGRAM ARP (INPUT=220,OUTPUT=220,TAPE5=INPUT,TAPE6=OUTPUT,       000100
     CTAPE1=220,TAPE2=220,TAPE3=220,TAPE4=220,TAPE8=220)                000110
      DIMENSION ANAM(50),DATA(50),PK1(40,20,8)                          000120
      DIMENSION PKS(12,6,10),PKM(50)                                    000130
      DIMENSION IKS(12,6,10),IKM(50),RGRD(8,41),DGRD(8,21)              000140
      DIMENSION HGT(9),XOMG(3),INEW(50),VTHT(5),GLTR(3,10)              000150
```

```
      DIMENSION ALPHA(13),BETA(7),PVT(5)
      DIMENSION PKG(10),CEP(10),RANGE(11),RSUM(10)
      DIMENSION PK(10),PKR(10),PKD(10),PKBL(10)
      DIMENSION RDH(5,2),DDH(5,2),HDH(5,2)
      DIMENSION RRG(10),RRSG(10),BRG(10),BRSG(10),BDG(10),BDSG(10)
      DIMENSION BHG(10),BHSG(10)
      COMMON /SRCH/IPEN,IPN,RBS,DBS,HBS,BR,BD,BH,OMEGA,RPN
     1,DPN,HPN,RDH1,RDH2,DDH1,DDH2,HDH1,HDH2
      COMMON /RDWRT/HGT,XOMG,VTHT,GLTR,PVT,GMR,GMD,
     1 GMH,SDD(10),SDH(10),IDAT(10),RDR(5),BBLST(5),HBLST(5),RU,DU
      DATA ALPHA/0.,30.,60.,90.,120.,150.,180.,210.,240.,270.,300.,
     C330.,360./
      DATA BETA/0.,15.,30.,45.,60.,75.,90./
      DATA DATA/50*0./,YES/1HY/,INEW/50*0/,GLTR/30*0./
      DATA ANAM/4HFZAM,4HFZTM,4HPKDH,4HPKBL,4HFZAS,4HFZTS,4HOMEG,
     C4HNGER,4HNCEP,4HFUNC,4HDHIT,4HPHIM,4HPHIS,4HSAMP,4HRADR,
     C4HDHAZ,4HPKNH,4HPKNA,4HOMGS,4HPDVT,4HPKPF,4HGLTR,4HSRNG,4HPRNT,
     C4HDBUG,4HTGTC,4HDUDR,4HBLST,22*4H    /
      DATA IDAT/8,11,15,17,18,20,22,28,2*0/
      DATA END/4HEND /
      CALL CONNEC (5LTAPE5)
      CALL CONNEC (5LTAPE6)
      REWIND 8
      WRITE (6,2002)
C
      WRITE (6,*) "     ANTI-RADIATION SIMULATION PROGRAM - 1/30/81"
C
      WRITE (6,2002)
      WRITE (6,*) "    NOTE:  ALL COORDINATES ARE DEFINED RELATIVE TO"
      WRITE (6,*) "           ORIGIN AT GROUND ZERO OF TARGET."
C
      WRITE (6,*) "           COORDINATE SYSTEM IS RECTANGULAR,"
      WRITE (6,*) "           TARGET HEADING IS NEGATIVE RANGE,"
      WRITE (6,*) "           DRIVER SIDE (L) IS POSITIVE DEFLECTION,"
      WRITE (6,*) "           HEIGHT IS MEASURED FROM GROUND."
C
C          INEW = 1, NEW DATA THIS CASE.
C
      WRITE (6,2002)
      CALL DATE (DTE)
      CALL TIME (TME)
      WRITE (6,2005) DTE,TME
 2005 FORMAT (//,5X,*DATE - *,A10,/,5X,*TIME - *,A10)
 2002 FORMAT (//,50("*"),/)
 2001 FORMAT (///)
      WRITE (6,2002)
      WRITE (6,*) "DO YOU WANT A LISTING OF CODE NAMES? "
      READ (5,1001) ANS
      IF(ANS.NE.YES) GO TO 54
      WRITE (6,2002)
      WRITE (6,*) "OMEG - ELEVATION OF ATTACK ANGLE"
```

```
      WRITE (6,*) "  NOTE:  FOLLOWING GUIDANCE ERROR PARAMETERS"       000670
      WRITE (6,*) "         (SIGD,SIGH) ARE MEASURED"                  000680
      WRITE (6,*) "         IN PLANE NORMAL TO TRAJECTORY AND"         000690
      WRITE (6,*) "         PASSING THROUGH HOMING POINT"              000700
      WRITE (6,*) "NGER - NUMBER OF GUIDANCE ERRORS TO CONSIDER"       000710
      WRITE (6,*) "         ENTER HOMING POINT (R,D,H), GUIDANCE"      000720
      WRITE (6,*) "         ERRORS ARE DISTRIBUTED ABOUT HOMING PT."   000730
      WRITE (6,*) "NCEP - 1., IF CEP IS INPUT FOR GUIDANCE ERROR SIGMAS"000740
      WRITE (6,*) "FZAM,FZAS,FZTM,FZTS - FUZING ERROR OPTIONS"         000750
      WRITE (6,*) "  FZAM - MEAN ANGLE AT WHICH FUZING OCCURS ON"      000760
      WRITE (6,*) "          INTERCEPT"                                000770
      WRITE (6,*) "  FZAS - STD DEV ASSOCIATED WITH FZAM"              000780
      WRITE (6,*) "NOTE:  FUZE ANGLE IS CONSTRAINED TO (0,PI)"         000790
      WRITE (6,*) "NOTE:  FOR UNIFORM FUZING ANGLE BETWEEN FZAM"       000800
      WRITE (6,*) "       AND FZAS, ENTER A NEGATIVE VALUE FOR FZAM"   000810
      WRITE (6,*) "       FUZE ANGLE WILL BE CHOSEN UNIFORMLY RANDOM"  000820
      WRITE (6,*) "       BETWEEN POSITIVE FZAM AND FZAS"              000830
      WRITE (6,*) "       FOR TIME-TO-GO FUZE, ENTER NEGATIVE FZAS."   000840
      WRITE (6,*) "NOTE:  FUZING PLANE PASSES THROUGH FUZING GLITTER"  000850
      WRITE (6,*) "       POINT NORMAL TO SAMPLE TRAJECTORY"           000860
      WRITE (6,*) "  FZTM - MEAN DISTANCE FROM GUIDANCE PLANE AT WHICH" 000870
      WRITE (6,*) "          FUZING WILL OCCUR ALONG TRAJECTORY"       000880
      WRITE (6,*) "NOTE:  ENTER A NEGATIVE FZTM FOR HEIGHT FUZING "    000890
      WRITE (6,*) "       WITH MEAN HEIGHT ABS(FZTM) "                 000900
      WRITE (6,*) "  FZTS - STD DEV ASSOCIATED WITH FZTM"              000910
      WRITE (6,*) "SAMP - SAMPLE SIZE"                                 000920
      WRITE (6,*) "FKNH - NUMBER OF HEIGHTS AT WHICH FRAGMENTATION"    000930
      WRITE (6,*) "       PK DATA WILL BE DEFINED"                     000940
      WRITE (6,*) "       NOTE: FKNH < 9"                              000950
      WRITE (6,*) "PKPF - PROBABILITY OF PRIMARY FUZE FUNCTIONING"     000960
      WRITE (6,*) "PDVT - 0. FOR PD BACKUP, NVT FOR VT BACKUP FUZE"    000970
      WRITE (6,*) "       WHERE NVT = NUMBER OF VT BURST HEIGHTS"      000980
      WRITE (6,*) "GLTR - 0. IF PRIMARY FUZE FUNCTIONS RELATIVE TO"    000990
      WRITE (6,*) "       CENTER OF TARGET, NGLT IF PRIMARY FUZE"      001000
      WRITE (6,*) "       FUNCTIONS  RELATIVE TO ANY ONE OF NGLT"      001010
      WRITE (6,*) "       EQUALLY LIKELY GLITTER POINTS ."             001020
      WRITE (6,*) "       SET NGLT NEGATIVE TO PICK FIRST"             001030
      WRITE (6,*) "       POINT ENCOUNTERED."                          001040
      WRITE (6,*) "SRNG - MAXIMUM RANGE FOR COMPUTING PK VS RANGE"     001050
      WRITE (6,*) "PRNT - 1. TO PRINT SUMMARY ONLY, 0. OTHERWISE"      001060
      WRITE (6,*) "DBUG - 6. TO PRINTOUT PROGRAM DEBUGGING DATA"       001070
      WRITE (6,*) "       DBUG = 1, GUIDANCE & FUZING DATA"            001080
      WRITE (6,*) "       DBUG = 2, DIRECT HIT PENETRATION DATA"       001090
      WRITE (6,*) "       DBUG = 4, PK BOX DATA"                       001100
      WRITE (6,*) "       DBUG = 5, PK GRIDS"                          001110
      WRITE (6,*) "       DBUG = 6, PK VS R DATA"                      001120
      WRITE (6,*) "TGTC - HEIGHT OF TARGET CENTER ABOVE GROUND"        001130
      WRITE (6,*) "DUDR - DUD RATE OF PROJECTILE, EXPRESSED AS A FRACTIO001140
     CN"                                                               001150
      WRITE (6,*) "DHIT - DIRECT HIT OPTION, NUMBER OF TARGET BOXES"   001160
      WRITE (6,*) "       IF DHIT IS OMITTED AND BLST IS INCLUDED,"    001170
      WRITE (6,*) "       BLST IS RADIUS FROM (0,0,TGTC) WITHIN"       001180
      WRITE (6,*) "       WHICH PKBLST = 1."                           001190
      WRITE (6,*) "PKDH - DIRECT HIT PK (0. = 1.)"                     001200
      WRITE (6,*) "PKBL - BLAST PK (0. = 1.)"                          001210
      WRITE (6,*) "RADR - 1., DEFINE FUNC FOR BLAST KILL OF RADAR ONLY" 001220
      WRITE (6,*) "       AND READ IN RADAR ANTENNA COORDINATES."      001230
```

```
      WRITE (6,*) "        TO DEFINE FUNC, SPECIFY R1 AND R2,"
      WRITE (6,*) "        WHERE BLAST PK IS 1 OUT TO R1 AND"
      WRITE (6,*) "        DECLINES LINEARLY TO 0 AT R2."
      WRITE (6,*) "DHAZ - AZIMUTH ANGLE OF ATTACK OFF FRONT OF TARGET"
      WRITE (6,*) "        TOWARD DRIVER SIDE.  SET TO -1. FOR RANDOM"
      WRITE (6,*) "BLST - BLAST RADIUS WITHIN WHICH VEHICLE PK=PKBL"
      WRITE (6,*) "NOTE:  TO ENTER BLAST RADII VS. BURST HEIGHT,"
      WRITE (6,*) "        ENTER NEGATIVE NUMBER OF BLAST,HGT PAIRS"
      WRITE (6,*) "        IN PLACE OF VALUE OF BLST.  PAIRS OF"
      WRITE (6,*) "        BLAST,HGT ARE ENTERED IN ASCENDING ORDER"
      WRITE (6,*) "        OF HEIGHT."
      WRITE (6,*) "COORDINATE SYSTEM IS RECTANGULAR, "
      WRITE (6,*) "TARGET HEADING IS NEGATIVE RANGE, "
      WRITE (6,*) "DRIVER SIDE (LEFT) IS POSITIVE DEFLECTION,"
      WRITE (6,*) "HEIGHT IS MEASURED FROM GROUND"
   54 NPRT = 0
      ISET = 0
      ITIME = 0
      CALL RDMOUT(INIT)
   15 CALL RDMIN(INIT)
      ISET = 1
      IF(IRD.EQ.5) GO TO 88
      IF(NPRT.GT.0) GO TO 80
      WRITE (6,*) "ENTER DATA BY ENTERING CODE NAME"
      WRITE (6,*) "FOLLOWED BY A COMMA AND THE VALUE IN FLOATING"
      WRITE (6,*) "POINT FORMAT.  TO END DATA ENTRY, ENTER "
      WRITE (6,*) "THE WORD END IN COLUMNS 1-3"
C
C         FILE TAPE1 CONTAINS BASIC INPUT DATA
C         FILES TAPE2 - TAPE4 CONTAIN FRAGMENTATION PK GRIDS
C         FOR DIFFERENT ANGLES OF ATTACK
C
   88 WRITE (6,*) "DO YOU WISH TO INITIALIZE DATA FROM"
      WRITE (6,*) "DATA FILE TAPE1?"
      READ (5,1001) ANS
      IRD = 5
      IF(ANS.EQ.YES) IRD = 1
   80 REWIND 1
      REWIND 2
      REWIND 3
      REWIND 4
      PI = ATAN2(0.,-1.)
      DO 51 I=1,10
   51 PKG(I) = 0.
C
C         INITIALIZE OR UPDATE DATA
C
      REWIND 1
    7 IF(IRD.EQ.5) WRITE (6,*) "ENTER DATA OR END - "
      READ (IRD,1000) AAAA,VALUE
 1000 FORMAT (A4,1X,F10.3)
      IF(AAAA.EQ.END) GO TO 14
      DO 53 J=1,50
      IF(AAAA.NE.ANAM(J)) GO TO 53
      INEW(J) = 1
      DATA(J) = VALUE
      GO TO 7
```

```
   53 CONTINUE
      WRITE (6,2000) AAAA
      GO TO 7
   14 CALL READ (DATA,INEW,ANAM,IRD,1,RDH,DDH,HDH)
C
C         SET UP TAPE1
C
    9 REWIND 1
      DO 81 I=1,50
      IF(DATA(I).EQ.0.) GO TO 81
      WRITE (1,1000) ANAM(I),DATA(I)
   81 CONTINUE
      WRITE (1,1000) END
      CALL WRITE (DAT..,1,CEP,RDH,DDH,HDH)
      REWIND 1
      IF(ITIME.EQ.0) GO TO 12
      WRITE (6 -' '"D YOU WANT CURRENT INPUT LISTED?"
      READ (5,100.) ANS
      IF(ANS.NE.YES) GO TO 23
      IF(ITIME.GT.0) WRITE (6,*) "CURRENT DATA - "
   12 IF(ITIME.EQ.0) WRITE (6,*) "INITIAL INPUTS - "
C
C         LIST DATA FILE (TAPE1)
C
      DO 8 I=1,50
      READ (1,1000) A,B
      IF(A.EQ.END) GO TO 6
    8 WRITE (6,1002) A,B
 1002 FORMAT (1X,A4,1X,F10.3)
    6 WRITE (6,1C 2) END
      CALL WRITE (DATA,6,CEP,RDH,DDH,HDH)
   23 REWIND 1
      ITIME = ITIME + 1
      IF(ISET.EQ.1) GO TO 89
      WRITE (6,*) "DO YOU WANT TO CHANGE ANY DATA? - "
      READ (5,1001) ANS
      IF(ANS.NE.YES) GO TO 82
   89 ISET = 0
C
C         READ IN CHANGES
C
      DO 13 I=1,50
   13 INEW(I) = 0
      DO 2 I=1,1000
      WRITE (6,*) "ENTER DATA OR END - "
      READ (5,1000) AAAA,VALUE
      IF(AAAA.EQ.END) GO TO 3
 1001 FORMAT (A1)
      DO 4 J=1,50
      IF(AAAA.NE.ANAM(J)) GO TO 4
      DATA(J) = VALUE
      INEW(J) = 1
      GO TO 2
    4 CONTINUE
      WRITE (6,2000) AAAA
 2000 FORMAT (1X,"***** DO NOT RECOGNIZE ",A4," *****")
    2 CONTINUE
```

PROGRAM ARP 73/74 OPT=1 FTN 4.8+508 03/13/81 05.39.35 PAGE 5

```
              3 CALL READ (DATA,INEW,ANAM,5,0,RDH,DDH,HDH)                002380
                GO TO 9                                                   002390
             82 DO 83 I=1,50                                              002400
             83 INEW(I) = 0                                               002410
          C                                                               002420
          C           SET UP DATA                                         002430
          C                                                               002440
          C           LOAD INPUT DATA INTO VARIABLE SET                   002450
          C           AND CONVERT DEGREES TO RADIANS                      002460
          C                                                               002470
                FZAM = DATA(1)/57.29578                                   002480
                FZTM = ABS(DATA(2))                                       002490
                PKDHX = DATA(3)                                           002500
                PKBLX = DATA(4)                                           002510
                FZAS = DATA(5` 57.29578                                   002520
                ITTG = 0                                                  002530
                IF(FZAS.LT.0.) ITTG = 1                                   002540
                FZAS = ABS(FZAS)                                          002550
                FZTS = DATA(6)                                            002560
                OMEG = DATA(7)/57.29578                                   002570
                NGER = DATA(8)                                            002580
                NCEP = DATA(9)                                            002590
                IFUN = 0                                                  002600
                NDHT = DATA(11)                                           002610
                NSMP = DATA(14)                                           002620
                NRDR = DATA(15)                                           002630
                DHAZ = DATA(16)/57.29578                                  002640
                NH = DATA(17)                                             002650
                NA = 0                                                    002660
                OMGS = 0.                                                 002670
                PKPF = DATA(21)                                           002680
                NVT = DATA(20)                                            002690
                NGLT = DATA(22)                                           002700
                JGLT = 1                                                  002710
                JGLT = ISIGN(JGLT,NGLT)                                   002720
                NGLT = IABS(NGLT)                                         002730
                SRNG = DATA(23)                                           002740
                NPRT = DATA(24)                                           002750
                NDSG = DATA(25)                                           002760
                TGTC = DATA(26)                                           002770
                DUDR = DATA(27)                                           002780
                BLST = DATA(28)                                           002790
                IF(BLST.LE.0.) GO TO 94                                   002800
                BBLST(1) = BLST                                           002810
                HBLST(1) - 100000.                                        002820
                BLST = 1.                                                 002830
             94 NBLST = ABS(BLST)                                         002840
                IHFZ = 0                                                  002950
                IF(DATA(2).LT.0.) IHFZ = 1                                002860
                IF(PKDHX.EQ.0.) PKDHX = 1.                                002870
                IF(PKBLX.EQ.0.) PKBLX = 1.                                002880
                NLOOP = NGER                                              002890
                IF(NDBG.GE.1) WRITE (6,*) "DBUG OPTION ",NDBG             002900
                IF(DATA(2).NE.0.) IFUZ = 2                                002910
                IF(DATA(1).NE.0.) IFUZ = 1                                002920
                XRNG = 0.                                                 002930
                IF(NDHT.EQ.0) GO TO 115                                   002340
```

```
      DO 116 I=1,NDHT
      RRNG = 10000.
      IF(SIGN(1.,DDH(I,1)).EQ.SIGN(1.,DDH(I,2))) GO TO 118
      IF(SIGN(1.,RDH(I,1)).EQ.SIGN(1.,RDH(I,2))) GO TO 116
      DO 119 J=1,2
      RRNG = AMIN1(RRNG,DDH(I,J))
  119 RRNG = AMIN1(RRNG,RDH(I,J))
      XRNG = SQRT(RRNG**2. + HDH(I,1)**2.)
      GO TO 116
  118 XRNG = HDH(I,1)
  116 CONTINUE
      XRNG = AMIN1(XRNG,HDH(NDHT,2))
  115 IF(SRNG.EQ.0.) SRNG = 100.
      DL = ALOG(SRNG-XRNG)/10.
      DO 111 I=1,10
      XI = I
  111 RANGE(I) = XRNG + EXP(DL*XI)
      RANGE(11) = 1000.
      IF(NVT.LE.1) GO TO 67
      DO 68 I=2,NVT
   68 PVT(I) = PVT(I) + PVT(I-1)
   67 IF(NGLT.GT.0) GO TO 59
      DO 60 I=1,3
   60 GLTR(I,1) = 0.
   59 IF(NA.EQ.0) GO TO 48
      DO 28 I=1,NA
   28 XOMG(I) = XOMG(I)/57.29578
   48 CONTINUE
C
C        READ IN PK GRIDS FOR EACH ATTACK ANGLE/BURST HEIGHT
C        COMBINATION
C
      IF(NH.EQ.0) GO TO 78
      CALL GRIDS (PK1,NH,2,RGRD,DGRD,NR,ND,NDBG)
C
C        LOOP OVER SIMULATIONS FOR EACH GUIDANCE ERROR SET
C
   78 DO 69 ILUP=1,NLOOP
C
C        INITIALIZE COUNTERS
C
      DO 70 I=1,50
      PKM(I) = 0.
   70 IKM(I) = 0
      DO 52 I=1,12
      DO 52 J=1,6
      DO 52 K=1,10
      IKS(I,J,K) = 0
   52 PKS(I,J,K) = 0.
      PKRADR = 0.
      PKDHIT = 0.
      PKBASE = 0.
      PKBLT = 0.
      PKTOT = 0.
      PKTOT2 = 0.
      RRBAR = 0.
      RRBAR2 = 0.
```

```
      BDBAR = 0.
      BDBAR2 = 0.
      BRBAR = 0.
      BRBAR2 = 0.
      BHBAR = 0.
      BHBAR2 = 0.
      IF(PKPF.EQ.0.) PKPF = 1.
      IF(PKPF.LT.0.) PKPF = 0.
      SIGD = SDD(ILUP)
      SIGH = SDH(ILUP)
      NCT = 0
C
C          BEGIN SIMULATIONS
C
      DO 1 ISIM=1,NSMP
      IF(DATA(16).LT.0.) DHAZ = RDM(1)*2.*PI
      PKSAMP = 0.0
      PKDH = 0.
      PKBLST = 0.
      PKRDR = 0.
C
C          CHECK FOR DUD
C
      IF(RDM(1).LE.DUDR) GO TO 16
C
C          SAMPLE FROM ATTACK ANGLE DISTRIBUTION
C
      CALL BOXNO (Z1,Z2)
      OMEGA = Z1*OMGS + OMEG
      SINO = SIN(OMEGA)
      COSO = COS(OMEGA)
      TANO = 1.
      IF(COSO.NE.0.) TANO = SINO/COSO
C
C          ROTATE COORDINATES OF HOMING POINT ACCORDING
C          TO AZIMUTH COMPONENT OF ATTACK ANGLE.
C
C          ALL COMPUTATIONS TO DETERMINE FUZING POINT ARE IN
C          ROTATED COORDINATE SYSTEM.
C
C
      GMRR = GMR
      GMDR = GMD
      CALL ROTATE (GMRR,GMDR,DHAZ,1.)
C
C          SAMPLE FROM GUIDANCE ERROR DISTRIBUTION
C          RELATIVE TO HOMING POINT
C
      CALL BOXNO (D,H)
      DMIN = SQRT((SIGH*H)**2. + (SIGD*D)**2.)
      GR = GMRR + SIGH*H*SINO
      GD = GMDR + SIGD*D
      GH = GMH + SIGH*H*COSO
C
C          (GR,GD,GH) IS INTERCEPT OF
C          TRAJECTORY WITH GUIDANCE PLANE
C          (RF,DF,HF) WILL BE FUZING POINT ON TRAJECTORY.
```

```
      C
            RF = GR
            DF = GD
            HF = GH
      C
      C          CHECK FOR PRIMARY FUZE FUNCTION
      C
            IBKUP = 0
            IF(RDM(1).GT.PKPF) GO TO 16
      C
      C          CHECK FOR HEIGHT FUZING
      C
            IF(IHFZ.EQ.1) GO TO 74
            O2 = 0.
      C
      C          CHECK FOR APPROPRIATE FUZING
      C
            CALL BOXNO (Z1,Z2)
            IGO = IFUZ + 1
            IF(NDBG.GE.1) WRITE (6,5003) IFUZ,IGO,GR,GD,GH
            GO TO (85,75,22,85),IGO
      C
      C          CHOSE GLITTER POINT FOR FUZING. ANGULAR FUZE ONLY
      C
         75 IF(JGLT.LT.0.AND.NGLT.GT.1) GO TO 76
            XGLT = NGLT
            IGLT = (RDM(1)-0.0001)*XGLT + 1.0
            IF(IGLT.EQ.0) IGLT = 1
         86 RGLT = GLTR(1,IGLT)
            DGLT = GLTR(2,IGLT)
            HGLT = GLTR(3,IGLT)
            IDO = 1
            GO TO 77
         76 IDO = NGLT
            GRMAX = -100000.
         77 DO 84 IGL=1,IDO
            IF(IDO.EQ.1) GO TO 21
            RGLT = GLTR(1,IGL)
            DGLT = GLTR(2,IGL)
            HGLT = GLTR(3,IGL)
         21 IF(NDBG.EQ.1) WRITE (6,*) *RGLT,DGLT,HGLT = *,RGLT,DGLT,HGLT
      C
      C          ROTATE GLITTER POINT INTO ARP COORDINATE SYSTEM
      C
            CALL ROTATE (RGLT,DGLT,DHAZ,1.)
            IF(NDBG.EQ.1) WRITE (6,*) *ROTATED GLITTER POINT = *
            IF(NDBG.EQ.1) WRITE (6,*) *DHAZ,RGLT,DGLT,HGLT = *,DHAZ,RGLT,DGLT,
           C HGLT
       5003 FORMAT (1X,*IFUZ,IGO = *,2(I2,*,*,1X),*GR,GD,GH = *,3(F6.1,*,*,1X)
           C)
      C
      C          USE LAW OF SINES AND LAW OF COSINES TO FIND
      C          FUZING POINT ON TRAJECTORY.  FIRST PICK A POINT
      C          ALONG TRAJECTORY TO COMPUTE BETAK (ANGLE BETWEEN
      C          TRAJECTORY AND A LINE (AB) FROM GLITTER POINT
      C          (RGLT,DGLT,HGLT) TO GUIDANCE PLANE INTERCEPT
      C          (GR,GD,GH) - NOTE THAT EVERYTHING IS IN ROTATED
```

```
C         COORDINATE SYSTEM (THROUGH AZIMUTH ATTACK ANGLE
C         COMPONENT).  THEN, KNOWING BETAX AND FUZING ANGLE
C         (ANG) COMPUTE ANGLE (GAMMA) WITH ITS VERTEX AT
C         GLITTER POINT AND OPPOSITE TRAJECTORY SEGMENT
C         BOUNDED BY GUIDANCE PLANE INTERCEPT AND FUZING
C         POINT.  FINALLY, KNOWING GAMMA, AB, AND ANG, COMPUTE
C         O2, THE DISTANCE FROM GUIDANCE PLANE INTERCEPT
C         TO FUZING POINT (USING THE LAW OF SINES).
C
      TANOX = TANO
      IF(SINO.EQ.0.) TANOX = 1.
      CB = 10.
      IF(SINO.NE.0.) CB = CB/SINO
C
C         GRL,GDL,GHL ARE COORDINATES OF A POINT ON
C         THE TRAJECTORY USED TO COMPUTE BETAX.
C
      GRL = GR - 10./TANOX
      GDL = GD
      GHL = GH
      IF(SINO.NE.0.) GHL = GH + 10.
      AB2 = (RGLT-GR)**2. + (DGLT-GD)**2. + (HGLT-GH)**2.
      BB2 = (RGLT-GRL)**2. + (DGLT-GDL)**2. + (HGLT-GHL)**2.
      AB = SQRT(AB2)
C
C         USE LAW OF COSINES TO COMPUTE BETAX, ANGLE WITH VERTEX AT
C         GLITTER POINT AND OPPOSITE TRAJECTORY SEGMENT
C         BOUNDED BY GUIDANCE PLANE INTERCEPT AND FUZING POINT.
C
      BETAX = ACOS((AB2-BB2+CB*CB)/(2.*AB*CB))
      IF(NDBG.EQ.1) WRITE (6,*) "BETAX,GRL,GDL,GHL,AB,CB = ",
     C BETAX,GRL,GDL,GHL,AB,CB
      FZASX = FZAS
      IF(ITTG.EQ.1) FZASX = 0.
C
C         ANGULAR FUZING FUNCTION
C
      ANG = Z2*FZASX + FZAM
      IF(FZAM.LT.0.) ANG = FZAM + RDM(1)*(FZASX-FZAM)
      IF(ANG.LT..01745) GO TO 18
      IF(ANG.GT.PI) GO TO 16
C
C         O2 IS DISTANCE ALONG TRAJECTORY FROM GUIDANCE
C         PLANE INTERCEPT TO FUZING POINT.
C
      GAMMA = PI - BETAX - ANG
C
C         IF GAMMA.LT.ZERO, USE SUPPLEMENT OF ANG FOR FUZING.
C
      IF(GAMMA.LT.0.) ANG = PI - ANG
      O2 = AB*(SIN(GAMMA)/SIN(ANG))
      IF(NDBG.EQ.1) WRITE (6,*) "O2,GAMMA,ANG = ",O2,GAMMA,ANG
      IF(IDO.EQ.1) GO TO 22
      IF(O2.LT.GRMAX) GO TO 84
      GRMAX = O2
      IGLT = IGL
   84 CONTINUE
```

```
      GO TO 86
C
C           LINEAR FUZING FUNCTION (ALONG TRAJECTORY)
C           FUZING DIRECTION IS POSITIVE IN THE NEGATIVE
C           RANGE DIRECTION, I.E., A POSITIVE CHANGE IN
C           THE FUZING DISTANCE, O2, IS IN THE NEGATIVE
C           RANGE DIRECTION.
C
   22 IF(ITTG.EQ.1) FZTS = DMIN*TAN(FZAS)
      O2 = O2 + Z2*FZTS + FZTM
      RF = GR - O2*COSO
      HF = GH + O2*SINO
      DF = GD
      GO TO 85
C
C           BACKUP FUZING
C
   16 HF = 0.
      IBKUP = 1
      IF(OMEGA.EQ.0.) GO TO  5
      IF(NVT.EQ.0) GO TO 17
   87 XK = RDM(1)
      DO 65 K=1,NVT
      KK = K
      IF(XK.LE.PVT(K)) GO TO 66
   65 CONTINUE
   66 HFX = VTHT(KK)
      IF(HFX.LE.HF) GO TO 24
      HF = HFX
   17 RF = GR - (HF-GH)/TANO
      DF = GD
      GO TO 61
    5 WRITE (6,*) "NO BACKUP FUZING FOR OMEGA = 0."
      WRITE (6,*) "TRAJECTORY CLOSEST POINT OF APPROACH TO TARGET"
      WRITE (6,*) "CENTER IS USED"
      RF = 0.
      DF = GD
      HF = GH
      GO TO 61
C
C           HEIGHT FUZING
C
   74 IF(SINO.EQ.0.) STOP 74
      CALL BOXNO (Z1,Z2)
      HF = FZTM + Z1*FZTS
      RF = RF + (GH-HF)/TANO
   85 IF(OMEGA.EQ.0.) GO TO 24
      IF(NVT.NE.0) GO TO 87
C
C           CHECK FOR FUZING POINT BELOW GROUND
C
   24 IF(HF.GE.0.) GO TO 61
      IF(OMEGA.EQ.0.) GO TO 61
      RF = RF + HF/TANO
      HF = 0.
C
C           PUT BURST POINT IN TARGET COORDINATE SYSTEM FOR
```

```
C         BLAST AND DIRECT HIT COMPUTATIONS.                           005800
C                                                                      005810
   61 CALL ROTATE (RF,DF,DHAZ,-1.)                                     005820
      BR = RF                                                          005830
      BD = DF                                                          005840
      BH = HF                                                          005850
      IF(NDBG.GE.1) WRITE (6,*) "BR,BD,BH AT STMT 61 = ",BR,BD,BH      005860
C                                                                      005870
C         SET UP BLST VALUE FOR BLST VS. HGT                           005880
C                                                                      005890
      IF(NBLST.LE.0) GO TO 105                                         005900
      DO 10 I=1,NBLST                                                  005910
      IF(HF.GT.HBLST(I)) GO TO 10                                      005920
      BLST = BBLST(I)                                                  005930
      GO TO 105                                                        005940
   10 CONTINUE                                                         005950
      BLST = 0.                                                        005960
      WRITE (6,*) "HF EXCEEDS ALL HBLST, HF = ",HF                     005970
      GO TO 18                                                         005980
  105 IF(NDHT.EQ.0) GO TO 106                                          005990
C                                                                      006000
C         DETERMINE DIRECT HIT PK                                      006010
C                                                                      006020
C                                                                      006030
C         USE 2 POINTS TO DEFINE TRAJECTORY, BURST POINT               006040
C         (BR,BD,BH) AND POINT AT BR+10 (RBS,DBS,HBS).                 006050
C         IF AZIMUTH ATTACK ANGLE IS 90 DEGREES, SET                   006060
C         RBS,DBS,HBS POINT AT BD+10.                                  006070
C         (RPN,DPN,HPN) WILL BE BURST POINT, WITH OR                   006080
C         WITHOUT DIRECT HIT.                                          006090
C                                                                      006100
C         IPN IS PENETRATION INDEX (0 = NO PENETRATION,                006110
C         N = BOX N PENETRATED)                                        006120
C                                                                      006130
      RPN = BR                                                         006140
      DPN = BD                                                         006150
      HPN = BH                                                         006160
      IF(ABS(DATA(16)).EQ.90.) GO TO 95                                006170
      RBS = BR + 10.                                                   006180
      DBS = BD - 10.*TAN(DHAZ)                                         006190
      HBS = BH - 10.*TANO/COS(DHAZ)                                    006200
      GO TO 96                                                         006210
   95 RBS = BR                                                         006220
      DBS = BD + 10.                                                   006230
      HBS = BH + 10.*TANO                                              006240
   96 IPN = 0                                                          006250
C                                                                      006260
C         CHECK EACH BOX FOR PENETRATION                               006270
C                                                                      006280
      IF(NDBG.EQ.1) WRITE (6,*) "OMEGA,RBS,DBS,HBS = ",OMEGA,RBS,DBS,HBS006290
      IF(NDBG.EQ.1) WRITE (6,*) "RF,DF,HF = ",RF,DF,HF                 006300
      IF(NDBG.EQ.1) WRITE (6,*) "GR,GD,GH = ",GR,GD,GH                 006310
      DO 92 I=1,NDHT                                                   006320
      IF(BR.LT.RDH(I,1)) GO TO 92                                      006330
      IF(DATA(16).NE.0.) GO TO 109                                     006340
      IF(BD.LT.DDH(I,1).OR.BD.GT.DDH(I,2)) GO TO 92                    006350
  109 IF(BH.GT.HDH(I,2).AND.OMEGA.GE.0.) GO TO 92                      006360
```

```
      IF(BH.LT.HDH(I,1).AND.OMEGA.EQ.0.) GO TO 92
      RDH1 = RDH(I,1)
      RDH2 = RDH(I,2)
      DDH1 = DDH(I,1)
      DDH2 = DDH(I,2)
      HDH1 = HDH(I,1)
      HDH2 = HDH(I,2)
C
C          IPEN = NUMBER OF SIDES PENETRATED (MUST BE 0 OR 2)
C
      IPEN = 0
      IF(ABS(DATA(16)).EQ.90.) GO TO 102
C
C          CHECK RANGE SIDES
C
      DO 97 K=1,2
      RDHX = RDH1
      IF(K.EQ.2) RDHX = RDH2
      CALL SEARCH (I,1,RDHX,DA,HA)
      IF(NDBG.EQ.2) WRITE (6,*) "IPEN,RDHX,DA,HA = ",IPEN,
     1 RDHX,DA,HA
   97 CONTINUE
      IF(IPEN.EQ.2) GO TO 92
  102 IF(DATA(16).EQ.0..OR.DATA(16).EQ.180.) GO TO 108
C
C          CHECK DEFLECTION SIDES
C
      DO 107 K=1,2
      DDHX = DDH1
      IF(K.EQ.2) DDHX = DDH2
      CALL SEARCH (I,2,RA,DDHX,HA)
      IF(NDBG.EQ.2) WRITE (6,*) "IPEN,RA,DDHX,HA = ",IPEN,
     1 RA,DDHX,HA
      IF(IPEN.EQ.2) GO TO 92
  107 CONTINUE
  108 IF(OMEGA.EQ.0.) GO TO 101
C
C          CHECK HEIGHT SIDES
C
      DO 117 K=1,2
      HDHX = HDH1
      IF(K.EQ.2) HDHX = HDH2
      CALL SEARCH (I,3,RA,DA,HDHX)
      IF(NDBG.EQ.2) WRITE (6,*) "IPEN,RA,DA,HDHX = ",IPEN,
     1 RA,DA,HDHX
      IF(IPEN.EQ.2) GO TO 92
  117 CONTINUE
  101 IF(IPEN.EQ.1) STOP 117
   92 CONTINUE
      IF(IPN.EQ.0) GO TO 106
      PKDH = PKDH + PKDHX
C
C          SET UP BURST COORDINATES (BR,BD,BH) FROM DIRECT HIT.
C
      BR = RPN
      BD = DPN
      BH = HPN
```

```
  106 IF(BH.GE.C.) GO TO 37
      IF(OMEGA.EQ.0.) STOP 106
      BR = BR + BH/TANO
      BH = 0.
C
C           COMPUTE NEAR MISS BLAST KILL
C
   37 IF(NBLST.EQ.0) GO TO 90
      IF(NDHT.EQ.0) GO TO 103
      DO 104 I=1,NDHT
      IBLST = 1
      CALL BLAST (IBLST,BR,BLST,RDH,I)
      CALL BLAST (IBLST,BD,BLST,DDH,I)
      CALL BLAST (IBLST,BH,BLST,HDH,I)
      IF(IBLST.EQ.1) GO TO 11
  104 CONTINUE
      GO TO 90
  103 DIST = SQRT(BR*BR + BD*BD + (BH-TGTC)*(BH-TGTC))
      IF(DIST.GT.BLST) GO TO 90
   11 PKBLST = PKBLST + PKBLX
C
C           COMPUTE RADAR BLAST KILL
C
   90 IF(NOBG.EQ.2) WRITE (6,*) "IPN,RPN,DPN,HPN,BR,BD,BH = ",
     C IPN,RPN,DPN,HPN,BR,BD,BH
      IF(NRDR.EQ.0) GO TO 27
      BRDR = BR-RDR(1)
      DRDR = BD-RDR(2)
      HRDR = BH-RDR(5)
      RRDR = SQRT(BRDR*BRDR+DRDR*DRDR+HRDR*HRDR)
      PKRDR = 1.0
      IF(RRDR.GT.RDR(4)) PKRDR = 1. - (RRDR-RDR(4))/(RDR(5)-RDR(4))
      IF(RRDR.GE.RDR(5)) PKRDR = 0.
 5004 FORMAT (1X,*BR,BD,BH = *,3(F6.1,*,*,1X))
   27 IBX = 0
      IROT = 0
      IF(NOBG.GE.1) WRITE (6,5004) BR,BD,BH
      IF(NH.EQ.0) GO TO 50
C
C           COMPUTE PK DUE TO FRAGMENTATION (PKSAMP)
C
C
C           INTERPOLATE IN RANGE, DEFLECTION, HEIGHT & ANGLE TO
C           GET FRAGMENTATION PK FROM PK GRIDS.
C
      II = 1
      IF(BH.GT.HG( NH+1)) GO TO 50
C
C           ROTATE BURST POINT FOR FRAGMENTATION PK
C           INTERPOLATION INTO ARP COORDINATE SYSTEM.
C           RECALL THAT PK GRIDS ARE IN PROJECTILE COORDINATE
C           SYSTEM.
C
      CALL ROTATE (BR,BD,CHAZ,1.)
      IROT = 1
C
C           LOCATE HEIGHT BOUNDARIES
```

```
C                                                                       007510
      DO 20 I=1,NH                                                      007520
      IH2 = I                                                           007530
      IF(BH.LE.HGT(I)) GO TO 25                                         007540
   20 CONTINUE                                                          007550
      IH2 = 0                                                           007560
   25 IH1 = IH2 - 1                                                     007570
      IF(IH1.EQ.0) IH1 = 1                                              007580
      IF(IH1.LT.0) IH1 = NH                                             007590
      IF(NDBG.EQ.4) WRITE (6,*) "IH1,IH2,NR,ND,RU,DU,BR,BD,BH = ",      007600
     C IH1,IH2,NR,ND,RU,DU,BR,BD,BH                                     007610
   31 CALL INTERP(BR,BD,BH,RGRD,DGRD,HGT,IH1,IH2,PK1,PKA,NR,ND,RU,DU,NH,007620
     C NDBG)                                                            007630
      PKSAMP = PKA                                                      007640
      GO TO 41                                                          007650
   50 PKSAMP = 0.                                                       007660
C                                                                       007670
C          COMPUTE SPHERICAL COORDINATES TO BURST POINT (BR,BD,BH)      007680
C          FROM GROUND ZERO  (0,0,0)                                    007690
C          SA1 = ANGLE OFF POSITIVE RANGE AXIS MEASURED                 007700
C          CLOCKWISE                                                    007710
C          SA2 = ANGLE OFF R-D PLANE MEASURED TOWARD POSITIVE           007720
C          SR = RANGE FROM BURST POINT TO (0,0,0)                       007730
C          H-AXIS IN VERTICLE PLANE                                     007740
C                                                                       007750
   41 IF(NDBG.EQ.4) WRITE (6,*) "PK(FRAG) = ",PKSAMP                    007760
C                                                                       007770
C          GET BURST POINT BACK INTO TARGET COORDINATE                  007780
C          SYSTEM IF IROT = 1.                                          007790
C                                                                       007800
      IF(IROT.EQ.1) CALL ROTATE (BR,BD,DHAZ,-1.)                        007810
      BRR = BR*BR                                                       007820
      BDD = BD*BD                                                       007830
      BHH = BH*BH                                                       007840
      RRR = BRR + BDD + BHH                                             007850
      RR = SQRT(RRR)                                                    007860
      WRITE (8,*) BR,BD,BH,RR                                           007870
      BRBAR = BRBAR + BR                                                007880
      BRBAR2 = BRBAR2 + BRR                                             007890
      BDBAR = BDBAR + BD                                                007900
      BDBAR2 = BDBAR2 + BDD                                             007910
      BHBAR = BHBAR + BH                                                007920
      BHBAR2 = BHBAR2 + BHH                                             007930
      RRBAR = RRBAR + RR                                                007940
      RRBAR2 = RRBAR2 + RRR                                             007950
      SA1 = PI/2.                                                       007960
      SA2 = 0.                                                          007970
      IF(BR.EQ.0.) GO TO 55                                             007980
      SA1 = ATAN2(BD,BR)                                                007990
      IF(SA1.LT.0.) SA1 = 2.*PI + SA1                                   008000
   55 IF(BD.EQ.0..AND.BR.EQ.0.) GO TO 56                                008010
      SA2 = ATAN(BH/SQRT(BR*BR+BD*BD))                                  008020
   56 SA1 = SA1*360./(2.*PI)                                            008030
      SA2 = SA2*360./(2.*PI)                                            008040
      DO 57 I=1,12                                                      008050
      ISA1 = I                                                          008060
      IF(SA1.LT.ALPHA(I+1)) GO TO 58                                    008070
```

```
   57 CONTINUE
   58 DO 98 I=1,6
      ISA2 = I
      IF(SA2.LT.BETA(I+1)) GO TO 99
   98 CONTINUE
   99 SR = SQRT(BR*BR + BD*BD + BH*BH)
      ISR = 0
      DO 100 I=1,10
      II = I
      IF(I.EQ.10) II = 11
      ISR = ISR + 1
      IF(SR.LT.RANGE(II)) GO TO 110
  100 CONTINUE
  110 IF(NDBG.EQ.6) WRITE (6,*) "ISA1,ISA2,ISR = ",ISA1,ISA2,ISR
      IF(NDBG.EQ.6) WRITE (6,*) "SA1,SA2,SR = ",SA1,SA2,SR
C
          STORE PK'S ACCORDING TO SPHERICAL COORDINATES
C
      IKS(ISA1,ISA2,ISR) = IKS(ISA1,ISA2,ISR) + 1
C
C         SUM PK'S OVER ALL SAMPLES
C
      IF(NDBG.GT.0) WRITE (6,*) "PKR,PKR,PKD,PKB = ",PKSAMP,PKRDR,PKDH
     C,PKBLST
      PKBASE = PKBASE + PKSAMP
      PKRADR = PKRADR + PKRDR
      PKDHIT = PKDHIT + PKDH
      PKBLT = PKBLT + PKBLST
      PKSAMP = 1. - (1.-PKSAMP)*(1.-PKRDR)*(1.-PKDH)*(1.-PKBLST)
      PKS(ISA1,ISA2,ISR) = PKS(ISA1,ISA2,ISR) + PKSAMP
      PKTOT = PKTOT + PKSAMP
      PKTOT2 = PKTOT2 + PKSAMP*PKSAMP
      IF(NDBG.GE.1) WRITE (6,3003) PKSAMP
 3003 FORMAT (5X,*SAMPLE PK = *,F6.4)
      IF(NPRT.EQ.1) GO TO 1
      IF(MOD(ISIM,10).NE.0) GO TO 1
      PKPRNT = ISIM
      PKPRNT = PKTOT/PKPRNT
      WRITE (6,*) "NO. SIMULATIONS, PK = ",ISIM,PKPRNT
      GO TO 1
   18 NCT = NCT + 1
    1 CONTINUE
C
C         DISPLAY FINAL RESULTS
C
      IF(NPRT.GT.0) GO TO 79
      WRITE (6,2002)
      WRITE (6,*) "FINAL RESULTS"
 3000 FORMAT (/,1X,*PK = *,F6.4,2X,*PKSD = *,F6.4,2X,*NSAMP = *,I6,/)
   79 XSAMP = NSMP
      PKBAR = PKTOT/XSAMP
      PKBASE = PKBASE/XSAMP
      PKRADR = PKRADR/XSAMP
      PKDHIT = PKDHIT/XSAMP
      PKBLT = PKBLT/XSAMP
      PK(ILUP) = PKBASE
      PKR(ILUP) = PKRADR
```

```
      PKD(ILUP) = PKDHIT
      PKG(ILUP) = PKBAR
      PKBL(ILUP) = PKBLT
      XSAMP = NSMP - NCT
      IF(NSMP.EQ.NCT) XSAMP = 1.
      XSMP = XSAMP-1.
      IF(XSMP.EQ.0.) XSMP = 1.
      BRG(ILUP) = BRBAR/XSAMP
      BDG(ILUP) = BDBAR/XSAMP
      BHG(ILUP) = BHBAR/XSAMP
      RRG(ILUP) = RRBAR/XSAMP
      BRSG(ILUP) = SQRT((BRBAR2 - XSAMP*BRG(ILUP)*BRG(ILUP))/XSMP)
      BDSG(ILUP) = SQRT((BDBAR2 - XSAMP*BDG(ILUP)*BDG(ILUP))/XSMP)
      BHSG(ILUP) = SQRT((BHBAR2 - XSAMP*BHG(ILUP)*BHG(ILUP))/XSMP)
      RRSG(ILUP) = SQRT((RRBAR2 - XSAMP*RRG(ILUP)*RRG(ILUP))/XSMP)
      IF(NCT.NE.0) WRITE (6,2004) NCT,NSMP,CEP(ILUP)
 2004 FORMAT (2X,*PROJECTILE OR FUZING DUDS = *,I4,* OUT OF *,I4,
     C * SIMULATIONS, *,/,2X,*GUIDANCE CEP = *,F6.2)
      IF(NPRT.GT.0) GO TO 69
      PKSD = (PKTOT2 - XSAMP*PKBAR*PKBAR)/XSMP
      IF(PKSD.LT.0.) PKSD = 0.
      PKSD = SQRT(PKSD)
      WRITE (6,3000) PKBAR,PKSD,NSMP
      WRITE (6,2002)
      WRITE (6,*) "DO YOU WANT PK VS R, ALPHA, BETA? "
      READ (5,1001) ANS
      IF(ANS.NE.YES) GO TO 44
C
C         PK VS R, ALPHA, BETA, WHERE ALPHA IS AZIMUTH ANGLE
C         MEASURED FROM POSITIVE RANGE AXIS TOWARD POSITIVE
C         DEFLECTION AXIS (0 TO 360), BETA IS ELEVATION ANGLE
C         MEASURED FROM NEGATIVE HEIGHT AXIS TO POSITIVE
C         HEIGHT AXIS (0 TO 90).
C
      WRITE (6,2001)
      WRITE (6,*) "  PK      R         ALPHA              BETA"
      WRITE (6,*) "------  -----  --------------  --------------"
      DO 49 I=1,10
      DO 49 J=1,12
      DO 49 K=1,6
      IF(IKS(J,K,I).EQ.0) GO TO 49
      XIKS = IKS(J,K,I)
      PKS(J,K,I) = PKS(J,K,I)/XIKS
   49 CONTINUE
      DO 45 I=1,10
      XI = 0.
      RSUM(I) = 0.
      RANG = RANGE(I)
      DO 47 J=1,12
      DO 47 K=1,6
      RPK = PKS(J,K,I)
      XIKS = IKS(J,K,I)
      XI = XI + XIKS
      RSUM(I) = RSUM(I) + XIKS*RPK
      IF(RPK.GT.0.) WRITE (6,3004) RPK,RANG,ALPHA(J),ALPHA(J+1),BETA(K),
     CBETA(K+1)
 3004 FORMAT (1X,F6.4,2X,F5.1,2(2X,F6.1,* - *,F6.1))
```

```
   47 CONTINUE                                                         009220
      IF(XI.EQ.0.) GO TO 45                                            009230
      RSUM(I) = RSUM(I)/XI                                             009240
   45 CONTINUE                                                         009250
      WRITE (6,2002)                                                   009260
      WRITE (6,*) "AVG PK VS. R"                                       009270
      WRITE (6,*) "------------"                                       009280
      DO 43 I=1,10                                                     009290
      R = RANGE(I)                                                     009300
      IF(RSUM(I).EQ.0.) GO TO 43                                       009310
      WRITE (6,3001) RSUM(I),R                                         009320
   43 CONTINUE                                                         009330
 3001 FORMAT (1X,F6.4,4X,F5.1)                                         009340
      WRITE (6,2002)                                                   009350
C          CHECK FOR ANOTHER CASE                                      009360
C                                                                      009370
   44 WRITE (6,2001)                                                   009380
   69 CONTINUE                                                         009390
C                                                                      009400
C          DISPLAY RESULTS FOR EACH GUIDANCE ERROR                     009410
C                                                                      009420
      IF(NPRT.GT.0) WRITE (6,2002)                                     009430
      FZTM = DATA(2)                                                   009440
      OMEGD = DATA(7)                                                  009450
      OMGSD = DATA(19)                                                 009460
      FZAMD = DATA(1)                                                  009470
      FZASD = DATA(5)                                                  009480
      WRITE (6,2006) OMEGD,OMGSD,FZAMD,FZASD,FZTM,FZTS,DHAZ,NSMP       009490
 2006 FORMAT (/,5X,*RESULTS FOR FOLLOWING CONDITIONS - *,//,           009500
     C12X,*ITEM*,13X,*MEAN*,4X,*STD DEV*,//,                          009510
     C10X,*ELEVATION*,4X,2F10.4,/,10X,*FUZE ANGLE*,3X,2F10.4,/,        009520
     C10X,*LINEAR FUZE*,2X,2F10.4,/,10X,*AZIMUTH      *,F10.4,/,       009530
     C10X,*SAMPLE SIZE - *,I5,/)                                       009540
      WRITE (6,2003) GMR,GMD,GMH                                       009550
      WRITE (6,2012)                                                   009560
 2012 FORMAT (/,5X,*ERROR DATA*,17X,*PK*,3X,                           009570
     C*PKFRAG PKRADR PKDHIT PKBLST*)                                   009580
 2003 FORMAT (5X,*HOMING POINT COORDINATES (R,D,H) = *,                009590
     C 2(F6.1,*, *),F6.1)                                              009600
      DO 72 I=1,NLOOP                                                  009610
      IF(NCEP.EQ.0) WRITE (6,2007) SDD(I),SDH(I),PKG(I)                009620
     C,PK(I),PKR(I),PKD(I),PKBL(I)                                     009630
      IF(NCEP.EQ.1) WRITE (6,2008) CEP(I),PKG(I)                       009640
     C,PK(I),PKR(I),PKD(I),PKBL(I)                                     009650
   72 CONTINUE                                                         009660
      WRITE (6,2002)                                                   009670
      WRITE (6,1003)                                                   009680
      DO 26 I=1,NLOOP                                                  009690
   26 WRITE (6,1004) CEP(I),RRG(I),RRSG(I),BRG(I),BRSG(I),BDG(I),BDSG(I)009700
     C ,BHG(I),BHSG(I)                                                 009710
 1003 FORMAT (/,5X,*BURST STATISTICS (MEAN, STD DEVIATION)*,/,1X,*CEP*, 009720
     C 4X,*BURST RANGE*,7X,*RANGE*,8X,*DEFLECTION*,7X,*HEIGHT*)        009730
 1004 FORMAT (1X,F4.1,4(2X,F6.2,1X,F6.2))                              009740
 2007 FORMAT (5X,*SD (D,H) - *,                                        009750
     C2(F4.1,*, *),1X,5F7.4)                                           009760
 2008 FORMAT (5X,*CEP - *,F4.1,                                        009770
     C14X,5F7.4)                                                       009780
```

```
 970              WRITE (6,*) "DO YOU WISH TO RUN ANOTHER CASE? "         009790
                  READ (5,1001) ANS                                       009800
                  IF(ANS.EQ.YES) GO TO 15                                 009810
                  STOP                                                    009820
                  END                                                     009830


     SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS     DEF LINE     REFERENCES
 3506   ARP            1

VARIABLES     SN   TYPE          RELOCATION
10232   A          REAL                          REFS      198      199   DEFINED      137
10227   AAAA       REAL                          REFS      166      168      173       218      221      226
                                              DEFINED      164      217
10373   AB         REAL                          REFS      486      487      507   DEFINED      480
10371   AB2        REAL                          REFS      480      486  DEFINED       478
31267   ALPHA      REAL      ARRAY               REFS        7      798    2*910   DEFINED       17
10455   ANAM       REAL      ARRAY               REFS        3      168      175       182      221      229
                                              DEFINED       21
10376   ANG        REAL                          REFS      496      497      502       506      507      508
                                              DEFINED      494      495      506
10217   ANS        REAL                          REFS       55      151      190       208      882      972
                                              DEFINED       54      149      169       207      881      971
10233   B          REAL                          REFS      199  DEFINED      197
  132   BBLST      REAL      ARRAY     RDWRT     REFS       15      584  DEFINED       272
10372   BB2        REAL                          REFS      486  DEFINED      479
    6   BD         REAL                SRCH      REFS       13      577      606       610      614    2*626      697
                                                2*702      708      712      721       738      751      753      772
                                                2*774      778      781      790       792    2*793    2*804
                                              DEFINED      575      683
10313   BDBAR      REAL                          REFS      781      864  DEFINED       343      781
10314   BDBAR2     REAL                          REFS      782      868  DEFINED       344      782
10426   BDD        REAL                          REFS      776      782  DEFINED       774
31542   BDG        REAL      ARRAY               REFS       11    2*868      961   DEFINED      864
31554   BDSG       REAL      ARRAY               REFS       11      961  DEFINED       868
31304   BETA       REAL      ARRAY               REFS        7      802    2*910   DEFINED       19
10374   BETAX      REAL                          REFS      487      502  DEFINED       486
    7   BH         REAL                SRCH      REFS       13      577      607       611      615      627      628
                                                  685      687      698    2*702       708      713      721      731
                                                  745      751      753    2*775       778      783      793    2*804
                                              DEFINED      576      684      688
10317   BHBAR      REAL                          REFS      783      865  DEFINED       347      783
10320   BHBAR2     REAL                          REFS      784      869  DEFINED       348      784
31566   BHG        REAL      ARRAY               REFS       12    2*869      961   DEFINED      865
10427   BHH        REAL                          REFS      776      784  DEFINED       775
31600   BHSG       REAL      ARRAY               REFS       12      961  DEFINED       869
10266   BLST       REAL                          REFS      271      272      275       696      697      698      703
                                              DEFINED      270      274      584       587
    5   BR         REAL                SRCH      REFS       13      577      605       609      613      624      687
                                                  696    2*702      708      711       721      738      751      753
                                                  772    2*773      778      779       789      790      792    2*793
                                                2*804  DEFINED      574      682       687
```

```
         PROGRAM ARP        73/74   OPT=1                      FTN 4.8+508       03/13/81  08.29.30       PAGE    19

VARIABLES      SN  TYPE          RELOCATION
10315  BRBAR       REAL                        REFS      779      863    DEFINED     345      779
10316  BRBAR2      REAL                        REFS      780      867    DEFINED     346      780
10413  BRDR        REAL                        REFS    2*714    DEFINED     711
31516  BRG         REAL       ARRAY            REFS       11    2*867       961    DEFINED     863
10425  BRR         REAL                        REFS      776      780    DEFINED     773
31530  BRSG        REAL       ARRAY            REFS       11      961    DEFINED     867
10365  CB          REAL                        REFS      469    3*486       487    DEFINED     468      469
31325  CEP         REAL       ARRAY            REFS        8      185       202     871        955      961
10334  COSD        REAL                        REFS    2*375      395       524    DEFINED     373
10340  D           REAL                        REFS      391      392       394
10404  DA          REAL                        REFS      646      647       670     671
10537  DATA        REAL       ARRAY            REFS        3      175       181     182        185      202      229
                                                239      240      241       242     243        247      248      249
                                                250      252      253       254     255        256      259      260
                                                261      265      266       267     268        269      270      277
                                                282      283      358       608     625        639    2*651      935
                                                936      937      938       939    DEFINED      20      170      222
    3  DBS         REAL              SRCH      REFS       13      620    DEFINED     610        614
31446  DDH         REAL       ARRAY            REFS       10      175       185     202        229    2*268      291
                                              2*626      631      632       697
10406  DDHX        REAL                        REFS      658      659    DEFINED     656        657
   16  DDH1        REAL              SRCH      REFS       13      656    DEFINED     631
   17  DDH2        REAL              SRCH      REFS       13      657    DEFINED     632
10347  DF          REAL                        REFS      573      575       621    DEFINED     402      526      544
                                                550
10357  DGLT        REAL                        REFS      440      444       446     478        479
                                             DEFINED     429      438
30735  DGRD        REAL       ARRAY            REFS        5      319       753
10252  DHAZ        REAL                        REFS      386      444       446     573        610      611      738
                                                772      940    DEFINED     255     358
10412  DIST        REAL                        REFS      703    DEFINED     702
10275  DL          REAL                        REFS      302    DEFINED     299
10342  DMIN        REAL                        REFS      522    DEFINED     392
   12  DPN         REAL              SRCH      REFS       13      683       708    DEFINED     606
10414  DRDR        REAL                        REFS    2*714    DEFINED     712
10215  DTE         REAL                        REFS       46       48
  145  DU          REAL              RDWRT     REFS       15      751       753
10265  DUDR        REAL                        REFS      366    DEFINED     269
 6522  END         REAL                        REFS      166      184       198     201        218
                                             DEFINED      26
10234  FZAM        REAL                        REFS      494    3*495    DEFINED     239
10453  FZAMD       REAL                        REFS      940    DEFINED     938
10240  FZAS        REAL                        REFS      245      246       489     522      DEFINED    243      246
10454  FZASD       REAL                        REFS      940    DEFINED     939
10375  FZASX       REAL                        REFS      494      495    DEFINED     489        490
10235  FZTM        REAL                        REFS      523      558       940    DEFINED     240      935
10242  FZTS        REAL                        REFS      523      558       940    DEFINED     247      522
10377  GAMMA       REAL                        REFS      506      507       508    DEFINED     502
10344  GD          REAL                        REFS      402      419       475     478        526      544      650
                                                622    DEFINED     394
10367  GDL         REAL                        REFS      479      487    DEFINED     475
10345  GH          REAL                        REFS      403      419       476     477        478      525      543
                                                551      559      622    DEFINED     395
10370  GHL         REAL                        REFS      479      487    DEFINED     476        477
   21  GLTR        REAL       ARRAY  RDWRT     REFS        6       15       428     429        430      437      438
                                                439    DEFINED     20       309
   65  GMD         REAL              RDWRT     REFS       15      385       946
```

```
VARIABLES      SN  TYPE                RELOCATION
10337  GMDR        REAL                           REFS        386       394   DEFINED       385
   66  GMH         REAL              RDWRT        REFS         15       395       946
   64  GMR         REAL              RDWRT        REFS         15       384       946
10336  GMRR        REAL                           REFS        386       393   DEFINED       384
10343  GR          REAL                           REFS        401       419       474       478       524       543       622
                                                DEFINED       393
10366  GRL         REAL                           REFS        479       487   DEFINED       474
10362  GRMAX       REAL                           REFS        510   DEFINED       434       511
10341  H           REAL                           REFS        391       392       393       395
10405  HA          REAL                           REFS        646       647       656       659
  137  HBLST       REAL       ARRAY  RDWRT        REFS         15       563   DEFINED       273
    4  HBS         REAL              SRCH         REFS         13       629   DEFINED       611       615
31460  HDH         REAL       ARRAY               REFS         10       175       185       202       229       293       295
                                                   297        627       628       633       634       698
10410  HDHX        REAL                           REFS        670       671   DEFINED       668       669
   20  HDH1        REAL              SRCH         REFS         13       668   DEFINED       633
   21  HDH2        REAL              SRCH         REFS         13       669   DEFINED       634
10350  HF          REAL                           REFS        541       543       559       565       567       576       583
                                                   588        621   DEFINED       403       525       531       542       551
                                                   558        568
10402  HFX         REAL                           REFS        541       542   DEFINED       540
10360  HGLT        REAL                           REFS        440       446       478       479   DEFINED       430       439
    0  HGT         REAL       ARRAY  RDWRT        REFS          6        15       731       745       753
   13  HPN         REAL              SRCH         REFS         13       684       708   DEFINED       607
10415  HRDR        REAL                           REFS      2*714   DEFINED       713
10326  I           INTEGER                        REFS        158       181     2*182       214       232     2*288     2*289
                                                   291        292       293       295       301       302     3*306       309
                                                 2*312        328       329       333       334       583       584       624
                                                 2*626        627       628       629       630       631       632       635
                                                   634        646       658       670       696       697       698       744
                                                   745        797       798       801       802       807       808       896
                                                   897      2*898       902       903       906       907     2*909     2*915
                                                   921        922       923     7*953     6*955     9*961
                                                DEFINED       157       180       196       213       215       231       286
                                                   300        305       308       311       327       330       582       623
                                                   694        743       796       800       806       893       909       920
                                                   952        960
10351  IBKUP       INTEGER                      DEFINED       407       532
10411  IBLST       INTEGER                        REFS        696       697       698       699   DEFINED       695
10417  IBX       * INTEGER                      DEFINED       719
  113  IDAT        INTEGER    ARRAY  RDWRT        REFS         15   DEFINED        25
10361  IDO         INTEGER                        REFS        435       436       509   DEFINED       431       433
10246  IFUN      * INTEGER                      DEFINED       251
10272  IFUZ        INTEGER                        REFS        418       419   DEFINED       282       283
10363  IGL         INTEGER                        REFS        437       438       439       512   DEFINED       435
10355  IGLT        INTEGER                        REFS        427       428       429       430   DEFINED       426       427
                                                   512
10353  IGO         INTEGER                        REFS        419       420   DEFINED       418
10270  IHFZ        INTEGER                        REFS        412   DEFINED       276       277
10423  IH1         INTEGER                        REFS        749       750       751       753   DEFINED       748       749
                                                   750
10422  IH2         INTEGER                        REFS        748       751       753   DEFINED       744       747
10421  II          INTEGER                        REFS        810   DEFINED       730       807       808
30143  IKM         INTEGER    ARRAY               REFS          5   DEFINED       329
26623  IKS         INTEGER    ARRAY               REFS          5       617       896       897       907
                                                DEFINED       333       617
10301  ILUP        INTEGER                        REFS        351       352       854       855       856       857       858
```

```
VARIABLES       SN  TYPE           RELOCATION
                                                863      864      865      866     3*867    3*868    3*869    3*870
                                                871  DEFINED      323
31205  INEW         INTEGER     ARRAY          REFS        6      175      229  DEFINED       20      169      214
                                                223      232
10223  INIT         INTEGER                    REFS      133      134
    0  IPEN         INTEGER           SRCH     REFS       13      647      650      659      661      671      673
                                                675  DEFINED      638
    1  IPN          INTEGER           SRCH     REFS       13      677      708  DEFINED      616
10224  IRD          INTEGER                    REFS      136      163      175  DEFINED      150      151
                                           I/O REFS      164
10420  IROT         INTEGER                    REFS      772  DEFINED      720      739
10434  ISA1         INTEGER                    REFS      812    2*817    2*828  DEFINED      797
10435  ISA2         INTEGER                    REFS      812    2*817    2*828  DEFINED      801
10221  ISET         INTEGER                    REFS      205  DEFINED      131      135      209
10324  ISIM         INTEGER                    REFS      834      835      837  DEFINED      357
10437  ISR          INTEGER                    REFS      809      812    2*817    2*828  DEFINED      805      809
10222  ITIME        INTEGER                    REFS      187      191      192      204  DEFINED      132      204
10241  ITTG         INTEGER                    REFS      490      522  DEFINED      244      245
10231  J            INTEGER                    REFS      168      169      170      221      222      223      291
                                                292      333      334      896      897    2*898      906      907
                                              2*910  DEFINED      167      220      290      331      894      904
10261  JGLT         INTEGER                    REFS      263      424  DEFINED      262      263
10302  K            INTEGER                    REFS      333      334      537      538      645      657      669
                                                896      897    2*898      906      907    2*910
                                            DEFINED      332      536      643      655      667      895      905
10401  KK           INTEGER                    REFS      540  DEFINED      537
10254  NA           INTEGER                    REFS      310      311  DEFINED      257
10267  NBLST        INTEGER                    REFS      581      582      692  DEFINED      275
10245  NCEP         INTEGER                    REFS      953      955  DEFINED      250
10323  NCT          INTEGER                    REFS      839      859      660    2*871  DEFINED      353      839
10300  ND           INTEGER                    REFS      319      751      753
10263  NDBG         INTEGER                    REFS    2*281      319      419      440      445      446      487
                                                508      577      620      621      622      647      659      671
                                                708      721      751      753      767      812      813      821
                                                831  DEFINED      267
10247  NDHT         INTEGER                    REFS      285      286      297      590      623      693      694
                                            DEFINED      252
10244  NGER         INTEGER                    REFS      280  DEFINED      249
10260  NGLT         INTEGER                    REFS      263      264      307      424      425      433
                                            DEFINED      261      264
10253  NH           INTEGER                    REFS      318      319      722      731      743      750      753
                                            DEFINED      256
10271  NLOOP        INTEGER                    REFS      323      952      960  DEFINED      280
10220  NPRT         INTEGER                    REFS      137      833      844      874      934
                                            DEFINED      130      266
10277  NR           INTEGER                    REFS      319      751      753
10251  NRDR         INTEGER                    REFS      710  DEFINED      254
10250  NSMP         INTEGER                    REFS      357      848      859      860      871      878      940
                                            DEFINED      253
10257  NVT          INTEGER                    REFS      304      305      534      536      561
                                            DEFINED      260
10243  OMEG         REAL                       REFS      371  DEFINED      248
   10  OMEGA        REAL              SRCH     REFS       13      372      373      533      560      566      620
                                                627      628      663      686  DEFINED      371
10451  OMEGD        REAL                       REFS      940  DEFINED      936
10255  OMGS         REAL                       REFS      371  DEFINED      258
10452  OMGSD        REAL                       REFS      940  DEFINED      937
```

```
VARIABLES      SN  TYPE          RELOCATION
10352  O2          REAL                          REFS      508      510      511      523      524      525
                                              DEFINED      413      507      523
10225  PI          REAL                          REFS      358      497      502      506      787      791      794
                                                  795  DEFINED      156
31364  PK          REAL      ARRAY               REFS        9      953      955  DEFINED      854
10424  PKA         REAL                          REFS      753      755
10442  PKBAR       REAL                          REFS      857    2*875      878  DEFINED      849
10305  PKBASE      REAL                          REFS      823      850      854  DEFINED      337      823      850
31422  PKBL        REAL      ARRAY               REFS        9      953      955  DEFINED      858
10327  PKBLST      REAL                          REFS      704      821      826      827  DEFINED      361      704
10306  PKBLT       REAL                          REFS      826      853      858  DEFINED      338      826      853
10237  PKBLX       REAL                          REFS      279      704  DEFINED      242      279
31410  PKD         REAL      ARRAY               REFS        9      953      955  DEFINED      856
10326  PKDH        REAL                          REFS      678      821      825      827  DEFINED      360      678
10304  PKDHIT      REAL                          REFS      825      852      856  DEFINED      336      825      852
10236  PKDHX       REAL                          REFS      278      678  DEFINED      241      278
31313  PKG         REAL      ARRAY               REFS        8      953      955  DEFINED      158      857
26541  PKM         REAL      ARRAY               REFS        4  DEFINED      328
10256  PKPF        REAL                          REFS      349      350      406  DEFINED      259      349      350
10440  PKPRNT      REAL                          REFS      836      837  DEFINED      835      836
31376  PKR         REAL      ARRAY               REFS        9      953      955  DEFINED      855
10303  PKRADR      REAL                          REFS      824      851      855  DEFINED      335      824      851
10330  PKRDR       REAL                          REFS      821      824      827  DEFINED      362      715      716
                                                  717
25221  PKS         REAL      ARRAY               REFS        4      828      898      906  DEFINED      334      828
                                                  898
10325  PKSAMP      REAL                          REFS      767      821      823      827      828      829    2*830
                                                  831  DEFINED      359      755      757      827
10444  PKSD        REAL                          REFS      876      877      878  DEFINED      875      876      877
10307  PKTOT       REAL                          REFS      829      836      849  DEFINED      339      829
10310  PKTOT2      REAL                          REFS      830      875  DEFINED      340      830
10621  PK1         REAL      ARRAY               REFS        3      319      753
   57  PVT         REAL      ARRAY    RDWRT      REFS        7       15    2*306      538  DEFINED      306
10450  R           REAL                          REFS      923  DEFINED      921
10407  RA          REAL                          REFS      658      659      670      671
10446  RANG        REAL                          REFS      910  DEFINED      903
31337  RANGE       REAL      ARRAY               REFS        8      810      903      921  DEFINED      302      303
    2  RBS         REAL               SRCH       REFS       13      620  DEFINED      609      613
31434  RDH         REAL      ARRAY               REFS       10      175      185      202      229    2*289      292
                                                  624      629      630      696
10403  RDHX        REAL                          REFS      646      647  DEFINED      644      645
   14  RDH1        REAL               SRCH       REFS       13      644  DEFINED      629
   15  RDH2        REAL               SRCH       REFS       13      645  DEFINED      630
  125  RDR         REAL      ARRAY    RDWRT      REFS       15      711      712      713    4*716      717
10346  RF          REAL                          REFS      559      567      573      574      621
                                              DEFINED      401      524      543      549      559      567
10356  RGLT        REAL                          REFS      440      444      446      478      479
                                              DEFINED      428      437
30225  RGRD        REAL      ARRAY               REFS        5      319      753
10447  RPK         REAL                          REFS      909    2*910  DEFINED      906
   11  RPN         REAL               SRCH       REFS       13      682      708  DEFINED      605
10431  RR          REAL                          REFS      778      785  DEFINED      777
10311  RRBAR       REAL                          REFS      785      866  DEFINED      341      785
10312  RRBAR2      REAL                          REFS      786      870  DEFINED      342      786
10416  RRDR        REAL                          REFS    2*716      717  DEFINED      714
31472  RRG         REAL      ARRAY               REFS       11    2*870      961  DEFINED      866
10274  RRNG        REAL                          REFS      291      292      293  DEFINED      287      291      292
```

```
           PROGRAM ARP         73/74    OPT=1                      FTN 4.8+508        03/13/81  08.29.30       PAGE     23

VARIABLES       SN  TYPE            RELOCATION
 10430   RRR        REAL                           REFS        777       786   DEFINED     776
 31504   RRSG       REAL        ARRAY              REFS         11       961   DEFINED     870
 31352   RSUM       REAL        ARRAY              REFS          8       909       915     922       923
                                                DEFINED        902       909       915
   144   RU         REAL                RDWRT      REFS         15       751       753
 10432   SA1        REAL                           REFS      2*791       794       798     813   DEFINED       787       790
                                                    791        794
 10433   SA2        REAL                           REFS        795       802       813   DEFINED       788       793       795
    67   SDD        REAL        ARRAY   RDWRT      REFS         15       351       953
   101   SDH        REAL        ARRAY   RDWRT      REFS         15       352       953
 10321   SIGD       REAL                           REFS        392       394   DEFINED     351
 10322   SIGH       REAL                           REFS        392       393       395   DEFINED       352
 10333   SINO       REAL                           REFS        375       393       467   2*469       477       525       550
                                                DEFINED        372
 10436   SR         REAL                           REFS        810       813   DEFINED     804
 10262   SRNG       REAL                           REFS        298       299   DEFINED     265       298
 10335   TANO       REAL                           REFS        466       543       559     567       611       615       687
                                                DEFINED        374       375
 10364   TANOX      REAL                           REFS        474   DEFINED       466     467
 10254   TGTC       REAL                           REFS      2*702   DEFINED       268
 10216   TME        REAL                           REFS         47        48
 10230   VALUE      REAL                           REFS        170       222   DEFINED     164       217
    14   VTHT       REAL        ARRAY   RDWRT      REFS          6        15       540
 10354   XGLT       REAL                           REFS        426   DEFINED       425
 10276   XI         REAL                           REFS        302       908       914     915   DEFINED       301       901
                                                    908
 10445   XIKS       REAL                           REFS        898       908       909   DEFINED       897       907
 10400   XK         REAL                           REFS        538   DEFINED       535
    11   XOMG       REAL        ARRAY   RDWRT      REFS          6        15       312   DEFINED       312
 10273   XRNG       REAL                           REFS        297       299       302   DEFINED       284       293       295
                                                    297
 10441   XSAMP      REAL                           REFS        849       850       851     852       853       861       863
                                                    864        865       866       867     868       869       870       875
                                                DEFINED        848       859       860
 10443   XSMP       REAL                           REFS        862       867       868     869       870       875
                                                DEFINED        861       862
  6521   YES        REAL                           REFS         55       151       190     208       882       972
                                                DEFINED         20
 10331   Z1         REAL                           REFS        370       371       417     557       558
 10332   Z2         REAL                           REFS        370       417       494     523       557

FILE NAMES          MODE
     0   INPUT
   410   OUTPUT
  1020   TAPE1      FMT                  WRITES        182       184     READS       197   MOTION        152       162       179
                                            186        203
  1430   TAPE2                           MOTION        153
  2040   TAPE3                           MOTION        154
  2450   TAPE4                           MOTION        155
     0   TAPE5      FMT                   READS         54       149       189       207     217       881       971
   410   TAPE6      MIXED                WRITES         30        32        34        35      36        38        39        40
                                             41         45        48        52        53      56        57        58        59
                                             60         61        62        63        64      65        66        67        68
                                             69         70        71        72        73      74        75        76        77
                                             78         79        80        81        82      83        84        85        86
                                             87         88        89        90        91      92        93        94        95
                                             96         97        98        99       100     101       102       103       104
```

```
FILE NAMES         MODE
                                  105     107     108     109     110     111     112     113     114
                                  115     116     117     118     119     120     121     122     123
                                  124     125     126     127     128     129     138     139     140
                                  141     147     148     163     173     188     191     192     199
                                  201     206     216     226     281     419     440     445     446
                                  487     508     546     547     548     577     588     620     621
                                  622     647     659     671     708     721     751     767     812
                                  813     821     831     837     845     846     871     878     879
                                  880     890     891     892     910     917     918     919     923
                                  926     929     934     940     946     947     953     955     958
                                  959     961     970
 3060  TAPE8      FREE         WRITES     778   MOTION     29
       VARIABLES USED AS FILE NAMES, SEE ABOVE

EXTERNALS        TYPE    ARGS        REFERENCES
       ACOS      REAL      1 LIBRARY    486
       ALOG      REAL      1 LIBRARY    299
       ATAN      REAL      1 LIBRARY    793
       ATAN2     REAL      2 LIBRARY    156     790
       BLAST               5            696     697     698
       BOXNO               2            370     391     417     557
       CONNEC              1             27      28
       COS       REAL      1 LIBRARY    373     611
       DATE                1             46
       EXP       REAL      1 LIBRARY    302
       GRIDS               8            319
       INTERP             16            753
       RDM       REAL      1            358     366     408     426     495     535
       RDMIN               1            134
       RDMOUT              1            133
       READ                8            175     229
       ROTATE              4            386     444     573     738     772
       SEARCH              5            646     658     670
       SIN       REAL      1 LIBRARY    372   2*507
       SQRT      REAL      1 LIBRARY    293     392     480     702     714     777     793     804     867
                                        868     869     870     877
       TAN       REAL      1 LIBRARY    522     610
       TIME                1             47
       WRITE               6            185     202

INLINE FUNCTIONS   TYPE    ARGS       DEF LINE  REFERENCES
       ABS       REAL      1  INTRIN            240     246     275     608     639
       AMIN1     REAL      0  INTRIN            291     292     297
       IABS      INTEGER   1  INTRIN            264
       ISIGN     INTEGER   2  INTRIN            263
       MOD       INTEGER   2  INTRIN            834
       SIGN      REAL      2  INTRIN          2*288   2*289

STATEMENT LABELS         DEF LINE   REFERENCES
 5725  1                   840        357     833     834     838
 4226  2                   228        215     224
 4231  3                   229        218
 4222  4                   225        220     221
 5103  5                   546        533
 4156  6                   201        198
 4057  7                   163        171     174
    0  8                   199        196
```

STATEMENT	LABELS		DEF LINE	REFERENCES			
4105	9		179	230			
5157	10		586	582	583		
5423	11		704	699			
4141	12		192	187			
0	13		214	213			
4103	14		175	166			
4005	15		134	972			
5053	16		531	408	497		
5075	17		543	534			
5723	18		839	366	496	589	
0	20		746	743			
4704	21		440	436			
5034	22		522	420	509		
4162	23		203	190			
5131	24		565	541	560		
5500	25		748	745			
0	26		961	960			
5454	27		719	710			
0	28		312	311			
0	31	INACTIVE	753				
5375	37		692	685			
5517	41		767	756			
6157	43		924	920	922		
6163	44		929	882			
6137	45		916	900	914		
0	47		913	904	905		
4466	48		313	310			
6065	49		899	893	894	895	896
5516	50		757	722	731		
0	51		158	157			
0	52		334	330	331	332	
4076	53		172	167	168		
4001	54		130	55			
5565	55		792	789			
5575	56		794	792			
0	57		799	796			
5610	58		800	798			
4456	59		310	307			
0	60		309	308			
5136	61		573	545	552	565	566
0	65		539	536			
5071	66		540	538			
4447	67		307	304			
0	68		306	305			
6165	69		930	323	674		
0	70		329	327			
0	72		957	952			
5115	74		556	412			
4645	75		424	420			
4671	76		433	424			
4674	77		435	432			
4471	78		323	318			
5735	79		848	844			
4035	80		152	137			
4117	81		183	180	181		
4234	82		231	208			
0	83		232	231			

```
            PROGRAM ARP        73/74   OPT=1                          FTN 4.8+508        03/13/81  08.29.30         PAGE     26

STATEMENT LABELS              DEF LINE   REFERENCES
 5031   84                       513       435      510
 5127   85                       550      2-420     527
 4662   86                       420       514
 5057   87                       535       561
 4023   88                       147       136
 4175   89                       209       205
 5425   90                       708       692      701      703
 5354   92                       676       623      624      626      627      628      650      661      673
 4324   94                       275       271
 5207   95                       613       608
 5215   96                       616       612
    0   97                       649       643
    0   98                       803       800
 5617   99                       804       802
    0  100                       811       806
 5351  101                       675       663
 5304  102                       651       639
 5414  103                       702       693
    0  104                       700       694
 5165  105                       590       581      585
 5366  106                       685       590      677
    0  107                       662       655
 5330  108                       663       651
 5242  109                       627       625
 5641  110                       812       810
    0  111                       302       300
 4415  115                       298       285
 4407  116                       296       286      294
    0  117                       674       657
 4495  118                       295       288      289
    0  119                       292       290
 7214  1000     FMT              165       174      182      184      197      217
 7320  1001     FMT              219        54      149      189      207      881      971
 7267  1002     FMT              200       199      201
10125  1003     FMT              953       959
10142  1004     FMT              965       961
 7326  2000     FMT              227       173      226
 6577  2001     FMT               51       890      929
 6574  2002     FMT               50        30       34       45       52       56      845      879      917      926
                                           934      958
10050  2003     FMT              950       946
 7650  2004     FMT              372       871
 6567  2005     FMT               49        48
10000  2006     FMT              941       940
10146  2007     FMT              966       953
10153  2008     FMT              968       955
10040  2012     FMT              948       947
 7333  3000     FMT              847       878
 7751  3001     FMT              925       923
 7614  3003     FMT              832       831
 7726  3004     FMT              912       910
 7370  5003     FMT              448       419
 7524  5004     FMT              713       721

LOOPS  LABEL    INDEX      FROM-TO     LENGTH    PROPERTIES
 4052  51        I         157 158       26       INSTACK
 4070  53        J         167 172      118         OPT       EXITS
```

```
           PROGRAM ARP        73/74    OPT=1                        FTN 4.8+508       03/13/81  08.29.30       PAGE    27

LOOPS   LABEL     INDEX      FROM-TO     LENGTH    PROPERTIES
 4110   81        I           180 183       12B                EXT REFS
 4145   8         I           196 199       11B                EXT REFS     EXITS
 4201   13        I           213 214        2B     INSTACK
 4205   2         I           215 228       24B                EXT REFS     EXITS    NOT INNER
 4214   4         J           220 225       11B       OPT        EXITS
 4237   83        I           231 232        2B     INSTACK
 4354   116       I           286 296       36B                EXT REFS  NOT INNER
 4372   119       J           290 292        6B     INSTACK
 4425   111       I           300 302        7B                EXT REFS
 4443   68        I           305 306        3B     INSTACK
 4453   60        I           308 309        2B     INSTACK
 4462   28        I           311 312        3B     INSTACK
 4472   69        ILUP        323 930     1476B                EXT REFS  NOT INNER
 4475   70        I           327 329        3B     INSTACK
 4502   52        I           330 334       20B                NOT INNER
 4503   52        J           331 334       15B                NOT INNER
 4511   52        K           332 334        3B     INSTACK
 4543   1         ISIM        357 840    11658B                EXT REFS  NOT INNER
 4675   84        IGL         435 513      137B                EXT REFS     EXITS
 5063   65        K           536 539        6B     INSTACK      EXITS
 5152   10        I           582 586       10B       OPT        EXITS
 5233   92        I           623 676      124B                EXT REFS  NOT INNER
 5265   97        K           643 649       15B                EXT REFS
 5311   107       K           655 662       17B                EXT REFS     EXITS
 5332   117       K           667 674       17B                EXT REFS     EXITS
 5400   104       I           694 700       14B                EXT REFS     EXITS
 5471   20        I           743 746        6B     INSTACK      EXITS
 5602   57        I           796 799        6B     INSTACK      EXITS
 5611   98        I           800 803        6B     INSTACK      EXITS
 5626   100       I           806 811       13B       OPT        EXITS
 6052   49        I           893 899       23B                NOT INNER
 6053   49        J           894 899       17B                NOT INNER
 6062   49        K           895 899        5B     INSTACK
 6076   45        I           900 916       44B                EXT REFS  NOT INNER
 6103   47        J           904 913       31B                EXT REFS  NOT INNER
 6104   47        K           905 913       26B                EXT REFS
 6150   43        I           920 924       12B                EXT REFS
 6211   72        I           952 957       40B                EXT REFS
 6255   26        I           960 961       24B                EXT REFS

COMMON BLOCKS   LENGTH
        SRCH        18
        RDWRT      102

STATISTICS
  PROGRAM LENGTH                 30053B    12331
  BUFFER LENGTH                   2707B     1479
  CM LABELED COMMON LENGTH         170B      120
          52000B CM USED
```

```
SUBROUTINE ROTATE       73/74    OPT=1                          FTN 4.8+508          03/13/81  08.29.30          PAGE    1

                 SUBROUTINE ROTATE (R,D,PHI,SIGNX)                                          009840
            C                                                                               009850
            C         ROTATES COORDINATE SYSTEM FROM TARGET SYSTEM                          009860
            C         TO PROJECTILE SYSTEM OR VICE VERSA, DEPENDING                         009870
            C         ON THE VALUE OF SIGNX (+1 = TO PROJECTILE SYSTEM,                     009880
            C         AND -1 = TO TARGET SYSTEM).                                           009890
            C                                                                               009900
                 RT = R                                                                     009910
                 R = R*COS(PHI) - SIGNX*D*SIN(PHI)                                          009920
                 D = D*COS(PHI) + SIGNX*RT*SIN(PHI)                                         009930
                 END                                                                        009940


SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS     DEF LINE     REFERENCES
    2  ROTATE        1            11

VARIABLES      SN   TYPE            RELOCATION
    0  D            REAL                 F.P.      REFS     9       10      DEFINED     1
    0  PHI          REAL                 F.P.      REFS    2*9     2*10     DEFINED     1        10
    0  R            REAL                 F.P.      REFS     8        9      DEFINED     1
   30  RT           REAL                           REFS    10    DEFINED      8
    0  SIGNX        REAL                 F.P.      REFS     9       10      DEFINED     1         9

EXTERNALS           TYPE    ARGS           REFERENCES
       COS          REAL      1 LIBRARY        9        10
       SIN          REAL      1 LIBRARY        9        10

STATISTICS
  PROGRAM LENGTH                      31B        25
       52000B CM USED
```

```
      SUBROUTINE READ (X,INEW,ANAM,IRD,IOPT,SR,SD,SH)                    009950
C                                                                        009960
C                                                                        009970
C         READ IN SUPPLEMENTAL INPUTS                                    009980
C                                                                        009990
      DIMENSION X(50),H(9),O(3),V(5),G(3,10),PV(5),ANAM(50),INEW(50)     010000
      DIMENSION SR(5,2),SD(5,2),SH(5,2)                                  010010
      COMMON /RDWRT/ H,O,V,G,PV,GMR,GMD,GMH,SDD(10),SDH(10),IDAT(10)     010020
     1, RDR(5),BBLST(5),HBLST(5),RU,DU                                   010030
      DO 11 ID=1,10                                                      010040
      IDR = IDAT(ID)                                                     010050
      IF(IDR.EQ.0.OR.(INEW(IDR).EQ.0.AND.IOPT.EQ.0)) GO TO 11            010060
      IF(X(28).GE.0..AND.ID.EQ.8) GO TO 11                               010070
      NN = ABS(X(IDR))                                                   010080
      IF(NN.EQ.0) GO TO 11                                               010090
      IF(IRD.EQ.5) WRITE (6,1009) ANAM(IDR)                              010100
      GO TO (1,2,7,3,4,5,6,8),ID                                         010110
    1 IF(X(9)) 21,20                                                     010120
   20 IF(IRD.EQ.5) WRITE (6,1007) NN                                     010130
      READ (IRD,*) (SDD(I),SDH(I),I=1,NN)                                010140
      GO TO 9                                                            010150
   21 IF(IRD.EQ.5) WRITE (6,1008) NN                                     010160
      READ (IRD,*) (SDD(I),I=1,NN)                                       010170
      DO 12 I=1,NN                                                       010180
      SDD(I) = SDD(I)/1.1774                                             010190
   12 SDH(I) = SDD(I)                                                    010200
    9 IF(IRD.EQ.5) WRITE (6,1012)                                        010210
      READ (IRD,*) GMR,GMD,GMH                                           010220
      GO TO 11                                                           010230
    2 IF(IRD.EQ.5) WRITE (6,1005) NN                                     010240
      READ (IRD,*) (SR(I,1),SR(I,2),SD(I,1),SD(1,2),SH(I,1),SH(I,2),I=1,010250
     C NN)                                                               010260
      GO TO 11                                                           010270
    3 NN = NN + 1                                                        010280
      IF(IRD.EQ.5) WRITE (6,1000) NN                                     010290
      READ (IRD,*) (H(I),I=1,NN)                                         010300
      IF(IRD.EQ.5) WRITE (6,1014)                                        010310
      READ (IRD,*) RU,DU                                                 010320
      GO TO 11                                                           010330
    4 IF(IRD.EQ.5) WRITE (6,1001) NN                                     010340
      READ (IRD,*) (O(I),I=1,NN)                                         010350
      GO TO 11                                                           010360
    5 IF(IRD.EQ.5) WRITE (6,1002) NN                                     010370
      READ (IRD,*) (V(I),I=1,NN)                                         010380
      IF(IRD.EQ.5) WRITE (6,1004) NN                                     010390
      READ (IRD,*) (PV(I),I=1,NN)                                        010400
      GO TO 11                                                           010410
    6 IF(IRD.EQ.5) WRITE (6,1003) NN                                     010420
      READ (IRD,*) ((G(I,J),I=1,3),J=1,NN)                               010430
      GO TO 11                                                           010440
    7 IF(IRD.EQ.5) WRITE (6,1006)                                        010450
      READ (IRD,*) (RDR(I),I=1,3)                                        010460
      IF(IRD.EQ.5) WRITE (6,1013)                                        010470
      READ (IRD,*) RDR(4),RDR(5)                                         010480
      GO TO 11                                                           010490
    8 IF(IRD.EQ.5) WRITE (6,1010) NN                                     010500
      IF(IRD.EQ.5) WRITE (6,1011)                                        010510
```

```
        READ (IRD,*) (BBLST(I),HBLST(I),I=1,NN)                            010520
     11 CONTINUE                                                           010530
   1000 FORMAT (1X,*ENTER *,I2,* HEIGHTS FOR FRAGMENTATION PK GRID,*,/,    010540
       C1X,*LAST VALUE CORRESPONDS TO HEIGHT*,/,                           010550
       C1X,*WHERE PK GOES TO ZERO.*)                                       010560
   1001 FORMAT (1X,*ENTER *,I2,* ELEVATION ANGLES ASSOC'TD WITH *,         010570
       C /,10X,*FRAGMENTATION PK DATA - *)                                 010580
   1002 FORMAT (1X,*ENTER *,I2,* VT FUZING HEIGHTS - *)                    010590
   1003 FORMAT (1X,*ENTER *,I2,* SETS OF GLITTER POINT COORDINATES (R,D,H)010600
       C - *)                                                              010610
   1004 FORMAT (1X,*ENTER *,I2,* PROB. VT DETONATION AT HEIGHT H - *)      010620
   1005 FORMAT (1X,*ENTER *,I2,* SETS OF BOUNDARIES FOR DIRECT HIT BOXES* 010630
       C,/,* ENTER FOR EACH BOX, MIN RANGE, MAX RANGE, MIN DEF, MAX DEF,* 010640
       C,/,* MIN HGT, MAX HGT - *)                                         010650
   1006 FORMAT (1X,*ENTER RADAR ANTENNA COORDINATES (R,D,H) RELATIVE*      010660
       C,/,1X,*TO TARGET GROUND ZERO. - *)                                 010670
   1007 FORMAT (1X,*ENTER *,I2,* SETS OF GUIDANCE ERRORS - *,              010680
       1/,3X,*STD DEV DEF, HGT - *,/)                                      010690
   1008 FORMAT (1X,*ENTER *,I2,* SETS OF GUIDANCE ERRORS - *,              010700
       1/,3X,*CEP - *,/)                                                   010710
   1009 FORMAT (1X,A4,1X,*DATA - *)                                        010720
   1010 FORMAT (1X,*ENTER *,I2,* SETS OF BLST, HGT DATA *)                 010730
   1011 FORMAT (1X,*BEGINNING WITH LOWEST HEIGHT - *)                      010740
   1012 FORMAT (1X,*ENTER COORDINATES OF HOMING POINT (R,D,H) - *,/)       010750
   1013 FORMAT (1X,*ENTER R1,R2, WHERE RADAR BLAST PK=1*,/,                010760
       C1X,*OUT TO R1 AND DECLINES LINEARLY*,/,                            010770
       C1X,*TO ZERO AT R2 - *)                                             010780
   1014 FORMAT (1X,*ENTER RANGE AND DEFLECTION DISTANCES*,/,               010790
       C1X,*FROM EDGE OF GRID TO WHERE THE FRAGMENTATION*,/,               010800
       C1X,*PK GOES TO ZERO - *)                                           010810
        END                                                                010820
```

SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS		DEF LINE	REFERENCES
3	READ	1	88

VARIABLES		SN	TYPE		RELOCATION									
0	ANAM		REAL	ARRAY	F.P.	REFS	6	16	DEFINED	1			
132	BBLST		REAL	ARRAY	RDWRT	REFS	8	DEFINED	58				
145	DU		REAL		RDWRT	REFS	8	DEFINED	38				
21	G		REAL	ARRAY	RDWRT	REFS	6	8	DEFINED	49			
65	GMD		REAL		RDWRT	REFS	8	DEFINED	28				
66	GMH		REAL		RDWRT	REFS	8	DEFINED	28				
64	GMR		REAL		RDWRT	REFS	8	DEFINED	28				
0	H		REAL	ARRAY	RDWRT	REFS	6	8	DEFINED	36			
137	HBLST		REAL	ARRAY	RDWRT	REFS	8	DEFINED	58				
711	I		INTEGER			REFS	2*20	23	2*25	2*26	6*31	36	41
						44	46	49	52	2*58	DEFINED	20	23
						24	31	36	41	44	46	49	52
						58							
706	ID		INTEGER			REFS	11	13	17	DEFINED	10		
113	IDAT		INTEGER	ARRAY	RDWRT	REFS	6	11					

```
VARIABLES        SN  TYPE             RELOCATION
  707  IDR           INTEGER                          REFS        2*12       14        16   DEFINED       11
    0  INEW          INTEGER     ARRAY       F.P.     REFS           6       12   DEFINED         1
    0  IOPT          INTEGER                 F.P.     REFS          12  DEFINED         1
    0  IRD           INTEGER                 F.P.     REFS          16       19        22        27       30       35       37
                                                        40          43       45        48        51       53       56       57
                                                   DEFINED           1  I/O REFS       20        23       28       31       36
                                                        38          41       44        46        49       52       54       58
  712  J             INTEGER                          REFS          49  DEFINED        49
  710  NN            INTEGER                          REFS          15       19        20        22       23       24       30
                                                        31          34       35        36        40       41       43       44
                                                        45          46       48        49        56       58
                                                   DEFINED          14       34
   11  O             REAL        ARRAY       RDWRT    REFS           6        8   DEFINED        41
   57  PV            REAL        ARRAY       RDWRT    REFS           6        8   DEFINED        46
  125  RDR           REAL        ARRAY       RDWRT    REFS           8  DEFINED        52      2*54
  144  RJ            REAL                    RDWRT    REFS           8  DEFINED        38
    0  SD            REAL        ARRAY       F.P.     REFS           7  DEFINED         1      2*31
   67  SDD           REAL        ARRAY       RDWRT    REFS           8       25        26   DEFINED       20       23       25
  101  SDH           REAL        ARRAY       RDWRT    REFS           8  DEFINED        20        26
    0  SH            REAL        ARRAY       F.P.     REFS           7  DEFINED         1      2*31
    0  SR            REAL        ARRAY       F.P.     REFS           7  DEFINED         1      2*31
   14  V             REAL        ARRAY       RDWRT    REFS           6        8   DEFINED        44
    0  X             REAL        ARRAY       F.P.     REFS           6       13        14        18  DEFINED        1

FILE NAMES           MODE
       TAPE6         FMT                   WRITES       16          19       22        27        30       35       37       40
                                               43       45          48       51        53        56       57
       VARIABLES USED AS FILE NAMES, SEE ABOVE

INLINE FUNCTIONS     TYPE    ARGS        DEF LINE  REFERENCES
       ABS           REAL       1  INTRIN                14

STATEMENT LABELS                 DEF LINE   REFERENCES
   46  1                             18          17
  117  2                             30          17
  145  3                             34          17
  166  4                             40          17
  200  5                             43          17
  223  6                             48          17
  236  7                             51          17
  253  8                             56          17
  110  9                             27          21
  277  11                            59          10       12          13       15        29        33       39       42       47
                                                 50       55
    0  12                            26          24
    0  20              INACTIVE      19          18
   70  21                            22          18
  476  1000        FMT               60          35
  514  1001        FMT               63          40
  526  1002        FMT               65          43
  534  1003        FMT               66          48
  544  1004        FMT               68          45
  553  1005        FMT               69          30
  574  1006        FMT               72          51
  606  1007        FMT               74          19
  617  1008        FMT               76          22
  627  1009        FMT               78          16
```

```
STATEMENT LABELS                   DEF LINE   REFERENCES
  632  1010     FMT                   79          56
  640  1011     FMT                   80          57
  645  1012     FMT                   81          27
  654  1013     FMT                   82          53
  670  1014     FMT                   85          37

LOOPS  LABEL    INDEX     FROM-TO     LENGTH    PROPERTIES
   7    11      ID        10 59        273B                EXT REFS  NOT INNER
  56            I         20 20         10B                EXT REFS
 104    12      I         24 26          3B      INSTACK
 126            I         31 31         15B                EXT REFS
 266            I         58 58         10B                EXT REFS

COMMON BLOCKS    LENGTH
      RDWRT        102

STATISTICS
  PROGRAM LENGTH                 747B      487
  CM LABELED COMMON LENGTH       146B      102
         52000B CM USED
```

SUBROUTINE WRITE 73/74 OPT=1 FTN 4.8+508 03/13/81 08.29.30 PAGE 1

```
      SUBROUTINE WRITE (X,IWRT,CEP,SR,SD,SH)                            010830
C                                                                       010840
C         WRITE LIST OF DATA (OUTPUT & TAPE1)                           010850
C                                                                       010860
C                                                                       010870
      DIMENSION X(50),H(9),O(3),V(5),G(3,10),PV(5),CEP(10)              010880
      DIMENSION SR(5,2),SD(5.2),SH(5,2)                                 010890
      COMMON /RDWRT/ H,O,V,G,PV,GMR,GMD,GMH,SDD(10),SDH(10),IDAT(10)    010900
     1, RDR(5),BBLST(5),HBLST(5),RU;DU                                  010910
      DO 8 I=1,50                                                       010920
      NN = ABS(X(I))                                                    010930
      IF(NN.EQ.0) GO TO 8                                               010940
      DO 10 J=1,10                                                      010950
      JJ = J                                                            010960
      IF(IDAT(JJ).EQ.I) GO TO 11                                        010970
   10 CONTINUE                                                          010980
      GO TO 8                                                           010990
   11 GO TO (1,2.7,3.4,5,6,9),JJ                                        011000
    1 IF(X(9)) 21,20                                                    011010
   20 WRITE (IWRT,*) (SDD(K),SDH(K),K=1,NN)                             011020
      GO TO 13                                                          011030
   21 DO 12 K=1,NN                                                      011040
   12 CEP(K) = SDD(K)*1.1774                                            011050
      WRITE (IWRT,*) (CEP(K),K=1,NN)                                    011060
   13 WRITE (IWRT,*) GMR,GMD,GMH                                        011070
      GO TO 8                                                           011080
    2 WRITE (IWRT,*) (SR(K,1),SR(K,2),SD(K,1),SD(K,2),SH(K,1),SH(K,2),  011090
     1 K=1,NN)                                                          011100
      GO TO 8                                                           011110
    3 NN = NN + 1                                                       011120
      WRITE (IWRT,*) (H(K),K=1,NN)                                      011130
      WRITE (IWRT,*) RU,DU                                              011140
      GO TO 8                                                           011150
    4 WRITE (IWRT,*) (O(K),K=1,NN)                                      011160
      GO TO 8                                                           011170
    5 WRITE (IWRT,*) (V(K),K=1,NN)                                      011180
      WRITE (IWRT,*) (PV(K),K=1,NN)                                     011190
      GO TO 8                                                           011200
    6 WRITE (IWRT,*) ((G(L,K),L=1,3),K=1,NN)                            011210
      GO TO 8                                                           011220
    7 WRITE (IWRT,*) (RDR(J),J=1,3)                                     011230
      WRITE (IWRT,*) RDR(4),RDR(5)                                      011240
      GO TO 8                                                           011250
    9 IF(X(I).GE.0.) GO TO 8                                            011260
      WRITE (IWRT,*) (BBLST(J),HBLST(J),J=1,NN)                         011270
    8 CONTINUE                                                          011280
      END                                                               011290
```

SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS	DEF LINE	REFERENCES
3 WRITE	1	47

```
      SUBROUTINE WRITE       73/74    OPT=1                            FTN 4.8+508         03/13/81  08.29.30        PAGE      2

 VARIABLES      SN  TYPE             RELOCATION
   132  BBLST        REAL       ARRAY    RDWRT          REFS         8         45
     0  CEP          REAL       ARRAY     F.P.          REFS         6         24    DEFINED         1         23
   145  DU           REAL                RDWRT          REFS         8         32
    21  G            REAL       ARRAY    RDWRT          REFS         6          6         39
    65  GMD          REAL                RDWRT          REFS         8         25
    66  GMH          REAL                RDWRT          REFS         8         25
    64  GMR          REAL                RDWRT          REFS         8         25
     0  H            REAL       ARRAY    RDWRT          REFS         6          8         31
   137  HBLST        REAL       ARRAY    RDWRT          REFS         8         45
   276  I            INTEGER                            REFS        11         15         44    DEFINED        16
   113  IDAT         INTEGER    ARRAY    RDWRT          REFS         8         15
     0  IWRT         INTEGER              F.P.       DEFINED         1   I/O REFS         20        24        25        27        31
                                                          32        34         36         37        39        41        42        45
   300  J            INTEGER                            REFS        14         41       2*45   DEFINED        13        41        45
   301  JJ           INTEGER                            REFS        15         18    DEFINED        14
   302  K            INTEGER                            REFS      2*20       2*23         24      6*27        31        34        36
                                                          37        39    DEFINED         20        22        24        27        31
                                                          34        36         37         39
   303  L            INTEGER                            REFS        39    DEFINED         39
   277  NN           INTEGER                            REFS        12         20         22        24        27        30        31
                                                          34        36         37         39        45   DEFINED        11        30
    11  O            REAL       ARRAY    RDWRT          REFS         6          8         34
    57  PV           REAL       ARRAY    RDWRT          REFS         6          8         37
   125  RDR          REAL       ARRAY    RDWRT          REFS         8         41       2*42
   144  RU           REAL                RDWRT          REFS         6         32
     0  SD           REAL       ARRAY     F.P.          REFS         7       2*27    DEFINED         1
    67  SDD          REAL       ARRAY    RDWRT          REFS         8         20         23
   101  SDH          REAL       ARRAY    RDWRT          REFS         8         20
     0  SH           REAL       ARRAY     F.P.          REFS         7       2*27    DEFINED         1
     0  SR           REAL       ARRAY     F.P.          REFS         7       2*27    DEFINED         1
    14  V            REAL       ARRAY    RDWRT          REFS         6          8         36
     0  X            REAL       ARRAY     F.P.          REFS         6         11         19        44   DEFINED         1
        VARIABLES USED AS FILE NAMES, SEE ABOVE

 INLINE FUNCTIONS    TYPE    ARGS         DEF LINE   REFERENCES
        ABS          REAL      1  INTRIN                 11

 STATEMENT LABELS               DEF LINE    REFERENCES
    27  1                          19          18
    74  2                          27          18
   116  3                          30          18
   127  4                          34          18
   135  5                          36          18
   150  6                          39          18
   157  7                          41          18
   202  8                          46          10        12        17        26        29        33        35        38        40
                                               43        44
   164  9                          44          18
     0  10                         16          13
    22  11                         18          15
     0  12                         23          22
    71  13                         25          21
     0  20         INACTIVE        20          19
    55  21                         22          19
```

```
LOOPS  LABEL     INDEX      FROM-TO     LENGTH    PROPERTIES
    7  8         I           10 46       176B               EXT REFS  NOT INNER
   14  10        J           13 16        53B     INSTACK     EXITS
   43            K           20 20       10B                EXT REFS
   60  12        K           22 23        3B      INSTACK
   77            K           27 27       15B                EXT REFS
  171            J           45 45       10B                EXT REFS

COMMON BLOCKS    LENGTH
       RDWRT       102

STATISTICS
  PROGRAM LENGTH                  320B      208
  CM LABELED COMMON LENGTH        146B      102
        52000B CM USED
```

```
      SUBROUTINE GRIDS (PK,NH,KK,R,D,NR,ND,NDBG)                          011300
C                                                                         011310
C          READ IN FRAGMENTATION PK GRID REDEFINE AND ORIENT              011320
C          AXES TO CORRESPOND WITH GEOMETRY OF MODEL                      011330
C                                                                         011340
C          GRIDS ARE IN ROTATED PROJECTILE COORDINATE SYSTEM.             011350
C                                                                         011360
C                                                                         011370
      DIMENSION PK(40,20,8),R(8,41),D(8,21)                               011380
      DO 1 I=1,NH                                                         011390
      READ (KK,1001) NR,ND                                                011400
      IF(NDBG.EQ.5) WRITE (6,2001) NR,ND                                  011410
      READ (KK,1000) (R(I,NR-J+1),J=1,NR)                                 011420
      READ (KK,1000) (D(I,ND-J+1),J=1,ND)                                 011430
      NR = NR-1                                                           011440
      ND = ND-1                                                           011450
C                                                                         011460
C          REDEFINE GRIDS AT CENTER OF CELLS (AT PK)                      011470
C          AND CHANGE SIGN OF GRID COORDINATES AND                        011480
C          CHANGE ALL INDICES TO GET GRID COORDINATES                     011490
C          IN ASCENDING ORDER AND IN PROPER RELATIONSHIP                  011500
C          TO ARPSIM GEOMETRY.                                            011510
C                                                                         011520
      DO 4 J=1,NR                                                         011530
    4 R(I,J) = -(R(I,J) + R(I,J+1))/2.                                    011540
      DO 5 J=1,ND                                                         011550
    5 D(I,J) = -(D(I,J) + D(I,J+1))/2.                                    011560
      IF(NDBG.EQ.5) WRITE (6,2000) (R(I,J),J=1,NR)                        011570
      IF(NDBG.EQ.5) WRITE (6,2000) (D(I,J),J=1,ND)                        011580
      DO 1 J=1,NR                                                         011590
    1 READ (KK,1002) (PK(NR-J+1,ND-K+1,I),K=1,ND)                         011600
      IF(NDBG.NE.5) RETURN                                                011610
      DO 2 I=1,NH                                                         011620
      DO 2 J=1,NR                                                         011630
      WRITE (6,2002) (PK(J,K,I),K=1,ND)                                   011640
    2 CONTINUE                                                            011650
 1000 FORMAT (10F7.1)                                                     011660
 1001 FORMAT (2I3)                                                        011670
 1002 FORMAT (10F7.5)                                                     011680
 2000 FORMAT (1X,10F7.1)                                                  011690
 2001 FORMAT (1X,2I3)                                                     011700
 2002 FORMAT (1X,10F7.5)                                                  011710
      END                                                                 011720
```

SYMBOLIC REFERENCE MAP (R=2)

```
ENTRY POINTS    DEF LINE    REFERENCES
   3  GRIDS         1          32         43

VARIABLES      SN  TYPE            RELOCATION
    0  D           REAL       ARRAY      F.P.      REFS        9      2*27       29   DEFINED       1        14        27
  307  I           INTEGER                         REFS       13        14     3*25      3*27       28        29        31
                                                     35    DEFINED        10       33
```

```
VARIABLES      SN  TYPE            RELOCATION
  310  J           INTEGER                             REFS       13        14      3*25      3*27        28        29        31
                                                         35    DEFINED      13        14        24        26        28        29
                                                         30       34
  311  K           INTEGER                             REFS       31        35    DEFINED       31        35
    0  KK          INTEGER               F.P.       DEFINED        1    I/O REFS      11        13        14        31
    0  ND          INTEGER               F.P.          REFS       12      2*14        16        26        29      2*31        35
                                                    DEFINED        1        11        16
    0  NDBG        INTEGER               F.P.          REFS       12        28        29        32    DEFINED        1
    0  NH          INTEGER               F.P.          REFS       10        33    DEFINED        1
    0  NR          INTEGER               F.P.          REFS       12      2*13        15        24        28        30        31
                                                         34    DEFINED       1        11        15
    0  PK          REAL         ARRAY    F.P.          REFS        9        35    DEFINED        1        31
    0  R           REAL         ARRAY    F.P.          REFS        9      2*25        28    DEFINED        1        13        25

FILE NAMES        MODE
      TAPE6       FMT                    WRITES        12        28        29        35
      VARIABLES USED AS FILE NAMES, SEE ABOVE

STATEMENT LABELS              DEF LINE    REFERENCES
    0  1                          31          10        30
    0  2                          36          33        34
    0  4                          25          24
    0  5                          27          26
  271  1000     FMT               37          13        14
  273  1001     FMT               38          11
  275  1002     FMT               39          31
  277  2000     FMT               40          28        29
  301  2001     FMT               41          12
  303  2002     FMT               42          35

LOOPS  LABEL     INDEX      FROM-TO     LENGTH     PROPERTIES
    7  1          I          10 31       150B                EXT REFS  NOT INNER
   20             J          13 13        11B                EXT REFS
   35             J          14 14        11B                EXT REFS
   56  4          J          24 25         4B       INSTACK
   70  5          J          26 27         4B       INSTACK
  101             J          28 28        10B                EXT REFS
  117             J          29 29        10B                EXT REFS
  131  1          J          30 31        24B                EXT REFS  NOT INNER
  134             K          31 31        15B                EXT REFS
  162  2          I          33 36        24B                EXT REFS  NOT INNER
  163  2          J          34 36        21B                EXT REFS  NOT INNER
  166             K          35 35        12B                EXT REFS

STATISTICS
  PROGRAM LENGTH                  350B       232
        52000B CM USED
```

```
SUBROUTINE SEARCH      73/74    OPT=1                          FTN 4.8+508          03/13/81  08.29.30        PAGE     1

 1                     SUBROUTINE SEARCH (II,JJ,R,D,H)                                              011730
             C                                                                                      011740
             C              DETERMINES WHETHER INTERCEPT OF TRAJECTORY WITH DIRECT                  011750
             C              HIT BOX PLANES FALLS WITHIN BOUNDARY OF DIRECT HIT BOX.                 011760
 5           C              UPDATES PENETRATION (BOX INTERCEPT) COORDINATES (RPN,DPN,HPN)011770
             C              IF HEIGHT COMPONENT INDICATES BOX PENETRATION OCCURS                    011780
             C              PRIOR TO CURRENT HPN POINT ALONG TRAJECTORY.  INITIAL VALUE             011790
             C              OF HPN IS BASED ON BURST POINT HEIGHT, BH.                              011800
             C                                                                                      011810
10                     COMMON /SRCH/ IPEN,IPN,RBS,DBS,HBS,BR,BD,BH,OMEGA,RPN,DPN                    011820
                      C,HPN,RDH1,RDH2,DDH1,DDH2,HDH1,HDH2                                           011830
                       DX(R1,R2,D1,D2,H1,H2) = (R2-R1)*(R2-R1)+(D2-D1)*(D2-D1)+(H2-H1)*(H011840
                      C2-H1)                                                                        011850
                       CALL INTRCP (R,D,H,JJ)                                                       011860
15                     IF(R.GT.RDH2.OR.R.LT.RDH1) RETURN                                            011870
                       IF(D.GT.DDH2.OR.D.LT.DDH1) RETURN                                            011880
                       IF(H.GT.HDH2.OR.H.LT.HDH1) RETURN                                            011890
                       IPEN = IPEN + 1                                                              011900
                       IF(OMEGA.GE.0..AND.H.LT.HPN) RETURN                                          011910
20                     IF(OMEGA.LT.0..AND.H.GT.HPN) RETURN                                          011920
                       RPN = R                                                                      011930
                       DPN = D                                                                      011940
                       HPN = H                                                                      011950
                       IPN = II                                                                     011960
25                     END                                                                          011970


   SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS      DEF LINE     REFERENCES
   3  SEARCH          1           15         16         17         19         20         25

VARIABLES       SN  TYPE            RELOCATION
   6  BD            REAL                  SRCH       REFS       10
   7  BH            REAL                  SRCH       REFS       10
   5  BR            REAL                  SRCH       REFS       10
   0  D             REAL                  F.P.       REFS       14      2*16        22    DEFINED          1
   3  DBS           REAL                  SRCH       REFS       10
  16  DDH1          REAL                  SRCH       REFS       10        16
  17  DDH2          REAL                  SRCH       REFS       10        16
  12  DPN           REAL                  SRCH       REFS       10     DEFINED      22
   0  H             REAL                  F.P.       REFS       14      2*17        19       20         23
                                                   DEFINED       1
   4  HBS           REAL                  SRCH       REFS       10
  20  HDH1          REAL                  SRCH       REFS       10        17
  21  HDH2          REAL                  SRCH       REFS       10        17
  13  HPN           REAL                  SRCH       REFS       10        19        20    DEFINED         23
   0  II            INTEGER               F.P.       REFS       24     DEFINED       1
   0  IPEN          INTEGER               SRCH       REFS       10        18     DEFINED      18
   1  IPN           INTEGER               SRCH       REFS       10     DEFINED      24
   0  JJ            INTEGER               F.P.       REFS       14     DEFINED       1
  10  OMEGA         REAL                  SRCH       REFS       10        19        20
   0  R             REAL                  F.P.       REFS       14      2*15        21    DEFINED          1
   2  RBS           REAL                  SRCH       REFS       10
```

```
VARIABLES     SN  TYPE            RELOCATION
   14  RDH1       REAL                 SRCH          REFS        10        15
   15  RDH2       REAL                 SRCH          REFS        10        15
   11  RPN        REAL                 SRCH          REFS        10     DEFINED        21

EXTERNALS           TYPE    ARGS          REFERENCES
       INTRCP                  4                  14

INLINE FUNCTIONS    TYPE    ARGS           DEF LINE  REFERENCES
       DX         REAL         6     SF           12

COMMON BLOCKS    LENGTH
       SRCH           18

STATISTICS
  PROGRAM LENGTH                    46B        38
  CM LABELED COMMON LENGTH          22B        18
        52000B CM USED
```

```
          SUBROUTINE INTRCP (R,D,H,IGO)                                  011980
C                                                                        011990
C               COMPUTES INTERCEPT OF TRAJECTORY WITH SIDES OF TARGET    012000
C               BOXES USING TWO POINTS. (BR,BD,BH) AND (RBS,DBS,HBS).    012010
C               INTERCEPT IS AT (R,D,H).                                 012020
C               SEE MAIN ROUTINE BETWEEN STATEMENTS 105 AND 96.          012030
C                                                                        012040
      COMMON /SRCH/ IPEN,IBS,RBS,DBS,HBS,BR,BD,BH,OMEGA,RPN,DPN          012050
     C,HPN,RDH1,RDH2,DDH1,DDH2,HDH1,HDH2                                 012060
      XOFY(GX,XA,Y,GY,YA) = GX + (XA-GX)*(Y-GY)/(YA-GY)                  012070
      GO TO (1,2,3),IGO                                                  012080
C                                                                        012090
C               GIVEN R, SOLVE FOR D,H                                   012100
C                                                                        012110
    1 D = XOFY(BD,DBS,R,BR,RBS)                                          012120
      H = XOFY(BH,HBS,R,BR,RBS)                                          012130
      RETURN                                                             012140
C                                                                        012150
C               GIVEN D, SOLVE FOR R,H                                   012160
C                                                                        012170
    2 R = XOFY(BR,RBS,D,BD,DBS)                                          012180
      H = XOFY(BH,HBS,D,BD,DBS)                                          012190
      RETURN                                                             012200
C                                                                        012210
C               GIVEN H, SOLVE FOR R,D                                   012220
C                                                                        012230
    3 R = XOFY(BR,RBS,H,BH,HBS)                                          012240
      D = XOFY(BD,DBS,H,BH,HBS)                                          012250
      RETURN                                                             012260
      END                                                                012270
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
11	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS	DEF LINE	REFERENCES		
3 INTRCP	1	17	23	29

VARIABLES	SN	TYPE	RELOCATION							
6 BD		REAL	SRCH	REFS	8	2*15	2*21	2*22	2*28	
7 BH		REAL	SRCH	REFS	8	2*16	2*22	2*27	2*28	
5 BR		REAL	SRCH	REFS	8	2*15	2*16	2*21	2*27	
0 D		REAL	F.P.	REFS	21	22	DEFINED	1	15	28
3 DBS		REAL	SRCH	REFS	8	15	21	22	28	
16 DDH1		REAL	SRCH	REFS	8					
17 DDH2		REAL	SRCH	REFS	8					
12 DPN		REAL	SRCH	REFS	8					
0 H		REAL	F.P.	REFS	27	28	DEFINED	1	18	22

```
VARIABLES      SN   TYPE              RELOCATION
    4  HBS          REAL                 SRCH        REFS         8      16      22      27      28
   20  HDH1         REAL                 SRCH        REFS         8
   21  HDH2         REAL                 SRCH        REFS         8
   13  HPN          REAL                 SRCH        REFS         8
    1  IBS          INTEGER              SRCH        REFS         8
    0  IGO          INTEGER              F.P.        REFS        11  DEFINED       1
    0  IPEN         INTEGER              SRCH        REFS         8
   10  OMEGA        REAL                 SRCH        REFS         8
    0  R            REAL                 F.P.        REFS        15      16 DEFINED       1      21      27
    2  RBS          REAL                 SRCH        REFS         8      15      16      21      27
   14  RDH1         REAL                 SRCH        REFS         8
   15  RDH2         REAL                 SRCH        REFS         8
   11  RPN          REAL                 SRCH        REFS         8

INLINE FUNCTIONS   TYPE    ARGS          DEF LINE  REFERENCES
      XOFY         REAL     5    SF        10         15      16      21      22      27      28

STATEMENT LABELS                 DEF LINE   REFERENCES
   15  1                            15          11
   27  2                            21          11
   41  3                            27          11

COMMON BLOCKS    LENGTH
      SRCH          18

STATISTICS
  PROGRAM LENGTH                    53B        43
  CM LABELED COMMON LENGTH          22B        18
         52000B CM USED
```

```
 1              SUBROUTINE BLAST (IB,P,B,X,I)                                  012280
          C                                                                    012290
          C          SET IB=0 IF BURST POINT IS OUT OF RANGE                   012300
          C          OF NEAR MISS BLAST                                        012310
 5        C                                                                    012320
                DIMENSION X(5,2)                                               012330
                IF(P.LT.(X(I,1)-B)) IB = 0                                     012340
                IF(P.GT.(X(I,2)+B)) IB = 0                                     012350
                END                                                            012360
```

```
        SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS      DEF LINE     REFERENCES
    3  BLAST          1            9

VARIABLES      SN  TYPE              RELOCATION
    0  B            REAL                   F.P.       REFS       7       8  DEFINED     1
    0  I            INTEGER                F.P.       REFS       7       8  DEFINED     1
    0  IB           INTEGER                F.P.    DEFINED       1       7     8
    0  P            REAL                   F.P.       REFS       7       8  DEFINED     1
    0  X            REAL        ARRAY      F.P.       REFS       6       7     8  DEFINED     1

STATISTICS
  PROGRAM LENGTH                 22B      18
         52000B CM USED
```

```
      SUBROUTINE INTERP (BR,BD,BH,RGRD,DGRD,HGT,IH1,IH2,PKS,PK,NR,ND,RU,
     C DU,NH,NDBG)
C
C         INTERPOLATES IN PK GRID TABLES.
C
      DIMENSION PKS(40,20,8),RGRD(8,41),DGRD(8,21),HGT(9)
      XINT(A,B,C,D,E) = E + (D-E)*(B-C)/(B-A)
C
C         FOR EACH HEIGHT, FIND R,D BOUNDS WHICH BRACKET BURST
C         POINT.
C
      P2 = -1.
C
C         INITIAL PASS FOR LOWER HEIGHT BOUND.
C
      IH = IH1
    4 CALL FIND (BR,RGRD,NR,IH,IR1,IR2)
      CALL FIND (BD,DGRD,ND,IH,ID1,ID2)
C
C         SET UP INTERPOLATION PARAMETERS & INTERPOLATE
C         TO GET APPROXIMATE PK(FRAG).
C
      R1 = -RU + RGRD(IH,1)
      IF(IR1.NE.0) R1 = RGRD(IH,IR1)
      R2 = RU + RGRD(IH,NR)
      IF(IR2.NE.0) R2 = RGRD(IH,IR2)
      D1 = -DU + DGRD(IH,1)
      IF(ID1.NE.0) D1 = DGRD(IH,ID1)
      D2 = DU + DGRD(IH,ND)
      IF(ID2.NE.0) D2 = DGRD(IH,ID2)
      IF(BR.LT.R1.OR.BR.GT.R2) GO TO 7
      IF(BD.LT.D1.OR.BD.GT.D2) GO TO 7
      IF(NDBG.EQ.4) WRITE (6,*) "IR1,IR2,ID1,ID2,R1,R2,D1,D2 = ",
     C IR1,IR2,ID1,ID2,R1,R2,D1,D2
C
C         INTERPOLTE FOR BURST RANGE ALONG LOWER DEFLECTION BOUND.
C
      PD1 = 0.
      IF(ID1.EQ.0) GO TO 1
      PR1 = 0.
      IF(IR1.NE.0) PR1 = PKS(IR1,ID1,IH)
      PR2 = 0.
      IF(IR2.NE.0) PR2 = PKS(IR2,ID1,IH)
      PD1 = XINT(R1,R2,BR,PR1,PR2)
      IF(NDBG.EQ.4) WRITE (6,*) "PR1,PR2,BR,PD1 = ",PR1,PR2,BR,PD1
    1 PD2 = 0.
C
C         INTERPOLATE FOR BURST RANGE ALONG UPPER DEFLECTION BOUND.
C
      IF(ID2.EQ.0) GO TO 2
      PR1 = 0.
      IF(IR1.NE.0) PR1 = PKS(IR1,ID2,IH)
      PR2 = 0.
      IF(IR2.NE.0) PR2 = PKS(IR2,ID2,IH)
      PD2 = XINT(R1,R2,BR,PR1,PR2)
      IF(NDBG.EQ.4) WRITE (6,*) "PR1,PR2,BR,PD2 = ",PR1,PR2,BR,PD2
C
```

```
      C          INTERPOLATE FOR BURST DEFLECTION ALONG BURST RANGE, LOWER
      C          HEIGHT.
      C
        2 IF(IH.EQ.IH1) P1 = XINT(D1,D2,BD,PD1,PD2)
          IF(NDBG.EQ.4) WRITE (6,*) "D1,D2,BD,P1 = ",D1,D2,BD,P1
          IF(IH1.EQ.IH2) GO TO 5
      C
      C          INTERPOLATE FOR BURST DEFLECTION ALONG BURST RANGE, UPPER
      C          HEIGHT.
      C
          IF(IH.EQ.IH2) P2 = XINT(D1,D2,BD,PD1,PD2)
          IF(P2.NE.-1.) GO TO 3
          IH = IH2
          IF(IH2.EQ.0) GO TO 6
      C
      C          REDO FOR UPPER HEIGHT BOUND.
      C
          GO TO 4
        6 P2 = 0.
          IH2 = NH + 1
      C
      C          INTERPOLATE FOR BURST HEIGHT.
      C
        3 PK = XINT(HGT(IH1),HGT(IH2),BH,P1,P2)
          RETURN
        5 PK = P1
          RETURN
        7 PK = 0.
          END
```

```
SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS      DEF LINE     REFERENCES
   0   INTERP        1           82          84         86

VARIABLES       SN  TYPE              RELOCATION
    0  BD           REAL                   F.P.    REFS        18      2*32       61        62        68
                                                   DEFINED      1
    0  BH           REAL                   F.P.    REFS        81    DEFINED       1
    0  BR           REAL                   F.P.    REFS        17      2*31       44        45        55        56
                                                   DEFINED      1
    0  DGRD         REAL        ARRAY      F.P.    REFS         6        18       27        28        29        30
                                                   DEFINED      1
    0  DU           REAL                   F.P.    REFS        27        29    DEFINED       1
  321  D1           REAL                           REFS        32        33       61        62        68
                                                   DEFINED     27        28
  322  D2           REAL                           REFS        32        33     2*61        62      2*68
                                                   DEFINED     29        30
    0  HGT          REAL        ARRAY      F.P.    REFS         6      3*81    DEFINED       1
  315  ID1          INTEGER                        REFS        18      2*28       33        39        41        43
  316  ID2          INTEGER                        REFS        18      2*30       33        50        53        54
  312  IH           INTEGER                        REFS        17        18       23        24        25        26        27
                                                     28        29        30       41        43        52        54        61
```

```
    SUBROUTINE INTERP      73/74    OPT=1                          FTN 4.3+508         03/13/81  08.29.30        PAGE      3

VARIABLES       SN  TYPE             RELOCATION
                                                  68     DEFINED        16       70
      0  IH1        INTEGER                F.P.     REFS        16       61       63        81    DEFINED         1
      0  IH2        INTEGER                F.P.     REFS        63       68       70        71       2*81
                                                 DEFINED         1       77
    313  IR1        INTEGER                         REFS        17     2*24       33      2*41       2*52
    314  IR2        INTEGER                         REFS        17     2*26       33      2*43       2*54
      0  ND         INTEGER                F.P.     REFS        18       29    DEFINED       1
      0  NDBG       INTEGER                F.P.     REFS        33       45       56        62    DEFINED         1
      0  NH         INTEGER                F.P.     REFS        77    DEFINED      1
      0  NR         INTEGER                F.P.     REFS        17       25    DEFINED       1
    323  PD1        REAL                            REFS        45       61       68    DEFINED        38        44
    326  PD2        REAL                            REFS        56     2*61     2*68    DEFINED        46        55
      0  PK         REAL                   F.P.  DEFINED         1       81       83        85
      0  PKS        REAL        ARRAY      F.P.     REFS         6       41       43        52         54
                                                 DEFINED         1
    324  PR1        REAL                            REFS        44       45       55        56    DEFINED        40        41
                                                    51          52
    325  PR2        REAL                            REFS      2*44       45     2*55        56    DEFINED        42        43
                                                    53          54
    327  P1         REAL                            REFS        62       81       83    DEFINED        61
    311  P2         REAL                            REFS        69     2*81    DEFINED      12         68        76
      0  RGRD       REAL        ARRAY      F.P.     REFS         6       17       23        24         25        26
                                                 DEFINED         1
      0  RU         REAL                   F.P.     REFS        23       25    DEFINED       1
    317  R1         REAL                            REFS        31       33       44        55    DEFINED        23        24
    320  R2         REAL                            REFS        31       33     2*44      2*55    DEFINED        25        26

FILE NAMES          MODE
       TAPE6        FREE                 WRITES       33        45       56       62

EXTERNALS              TYPE     ARGS          REFERENCES
       FIND                       6              17           18

INLINE FUNCTIONS       TYPE     ARGS          DEF LINE  REFERENCES
       XINT            REAL       5     SF       7          44        55       61       68       81

STATEMEN  LABELS                         DEF LINE   REFERENCES
    125   1                                  46        39
    155   2                                  61        50
    213   3                                  81        69
     10   4                                  17        75
    225   5                                  83        63
    210   6                                  76        71
    227   7                                  85        31          32

STATISTICS
  PROGRAM LENGTH                           345B       229
         52000B CM USED
```

```
 1              SUBROUTINE FIND (B,GRD,N,IH,IX1,IX2)                          013230
          C                                                                   013240
          C          FINDS BOUNDS OF BURST POINT IN PK GRID.                  013250
          C          IX1,IX2 ARE ARRAY ELEMENTS WHICH BRACKET                 013260
 5        C          SUBJECT COORDINATE.                                      013270
          C                                                                   013280
                DIMENSION GRD(8,41)                                           013290
                IF(B.LT.GRD(IH,1)) GO TO 1                                    013300
                IF(B.GT.GRD(IH,N)) GO TO 2                                    013310
10              DO 3 I=2,N                                                    013320
                IX2 = I                                                       013330
                IF(B.LT.GRD(IH,I)) GO TO 4                                    013340
              3 CONTINUE                                                      013350
              4 IX1 = IX2 - 1                                                 013360
15              RETURN                                                        013370
              1 IX2 = 1                                                       013380
                IX1 = 0                                                       013390
                RETURN                                                        013400
              2 IX2 = 0                                                       013410
20              IX1 = N                                                       013420
                END                                                           013430
```

```
      SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS      DEF LINE     REFERENCES
    3  FIND           1          15         18        21

VARIABLES      SN  TYPE          RELOCATION
    0  B           REAL                F.P.      REFS         8        9        12  DEFINED      1
    0  GRD         REAL       ARRAY    F.P.      REFS         7        8         9       12  DEFINED      1
   37  I           INTEGER                       REFS        11       12   DEFINED       10
    0  IH          INTEGER             F.P.      REFS         8        9        12  DEFINED      1
    0  IX1         INTEGER             F.P.    DEFINED        1       14        17       20
    0  IX2         INTEGER             F.P.      REFS        14   DEFINED        1       11      16      19
    0  N           INTEGER             F.P.      REFS         9       10        20  DEFINED      1

STATEMENT LABELS            DEF LINE   REFERENCES
   30  1                       16          8
   33  2                       19          9
    0  3                       13         10
   25  4                       14         12

LOOPS  LABEL    INDEX     FROM-TO    LENGTH    PROPERTIES
   16  3        I          10 13        7B     INSTACK    EXITS

STATISTICS
  PROGRAM LENGTH                46B      38
        52000B CM USED
```

DISTRIBUTION LIST

Commander
U.S. Army Armament Research
and Development Command
ATTN: DRDAR-LCS, R. Corn
DRDAR-LCU, A. Moss
DRDAR-LCE-M, A. King
DRDAR-LCU-CA, D. Costa
DRDAR-LCS-A, J. Brooks (6)
DRDAR-LCS-E, S. Einbinder
DRDAR-TSS (5)
DRDAR-GCL
Dover, NJ 07801

Project Manager
Cannon Artillery Weapons System
ATTN: DRCPM-CAWS-El, G. Waldron
Dover, NJ 07801

Director
U.S. Army Materiel Systems
Analysis Activity
ATTN: DRXSY-GS, J. Chernick
DRXSY-MP
Aberdeen Proving Ground, MD 21005

Commandant
U.S. Army Field Artillery School
ATTN: ATSF-CD-AD, R. Wilde
Fort Sill, OK 73503

Director
U.S. Army TRASANA
ATTN: ATAA-TBA, J. Hodde
White Sands Missile Range, NM 88002

Commander
Naval Surface Weapons Center
ATTN: Code G29, R. Scheibe
Dahlgren, VA 22448

Commander
U.S. Army Electronics Research
and Development Command
ATTN: DELHD-R-RSD, R. Parkhurst
Adelphi, MD 20783

Commander
Naval Weapons Center
ATTN: Code 3524, J. Netzer
China Lake, CA 93555

Administrator
Defense Technical Information Center
ATTN: Accessions Division (12)
Cameron Station
Alexandria, VA 22314

Commander/Director
Chemical Systems Laboratory
U.S. Army Armament Research
and Development Command
ATTN: DRDAR-CLJ-L
DRDAR-CLB-PA
APG, Edgewood Area, MD 21010

Director
Ballistics Research Laboratory
U.S. Army Armament Research
and Development Command
ATTN: DRDAR-TSB-S
Aberdeen Proving Ground, MD 21005

Chief
Benet Weapons Laboratory, LCWSL
U.S. Army Armament Research
and Development Command
ATTN: DRDAR-LCB-TL
Watervliet, NY 12189

Commander
U.S. Army Armament Materiel
Readiness Command
ATTN: DRSAR-LEP-L
Rock Island, IL 61299

Director
U.S. Army TRADOC Systems
Analysis Activity
ATTN: ATAA-SL
White Sands Missile Range, NM 88002